

UNIVERSITÉ DE SHERBROOKE
Faculté de génie
Département de génie électrique et de génie informatique

ÉTUDE DE PRÉCISION ET DE PERFORMANCE DU PROCESSUS DE CLASSIFICATION D'IMAGES DE PHYTOPLANKTON À L'AIDE DE MACHINES À VECTEURS DE SUPPORT

Mémoire de maîtrise
Spécialité : génie électrique

Eugène MORIN

Jury : Ruben GONZALEZ-RUBIO (directeur)
Daniel DALLE
Yannick HUOT

Sherbrooke (Québec) Canada

Mai 2014

À ma conjointe Penny et mes deux enfants,
Jeremy et Evelyn. Merci pour votre support
et vos encouragements.

RÉSUMÉ

Ce projet de recherche cible l'étude et l'amélioration de la précision de la classification d'images de phytoplancton et la diminution du temps de traitement moyen requis par image. Deux solutions de classification sont proposées pour atteindre ces objectifs. La première solution vise à effectuer la classification d'images en passant par les phases de prétraitement, de discrimination et de classification, et la deuxième solution utilise uniquement les phases de prétraitement et de classification.

En résumé, la phase de prétraitement manipule une image en vue de caractériser l'élément principal (le phytoplancton), la phase de discrimination utilise les arbres décisionnels à intervalles pour éliminer les catégories ayant peu ou pas de similitude avec l'image traitée et finalement, la phase de classification se sert de machines à vecteurs de support (SVM) pour prédire une catégorie d'appartenance à chaque image traitée.

À la base, il y a un appareil de capture automatisée d'images qui transmet celles-ci à un classificateur. Selon la vitesse de classification, une portion ou l'ensemble des images générées seront classifiés. Donc, plus le nombre d'échantillons à classifier est grand, meilleure est l'approximation de la population de chaque groupe de phytoplanctons, à un temps donné. Le but étant d'obtenir une analyse qualitative, quantitative et temporelle plus précise de ce micro-organisme.

Pour permettre la classification de ce type d'image, un logiciel nommé *Biotaxis* a été développé. Celui-ci offre à l'utilisateur l'option de choisir parmi les deux solutions de classification proposées ci-haut. Toutes deux débutent par l'entraînement d'un groupe de classification, qui est composé de plusieurs catégories d'image, suivi par des tests de classification, qui sont effectués sur ce groupe pour vérifier la précision de la classification des catégories d'image qui le compose. Pour entraîner et tester le classificateur du logiciel *Biotaxis*, deux ensembles d'images ont été employés. L'un d'eux sert uniquement à l'entraînement de groupes de classification et le second à tester ces derniers.

Les résultats obtenus dans ce projet de recherche ont permis de confirmer la validité des deux solutions proposées. Il fut possible d'atteindre une précision de la classification moyenne de 87 % et plus avec des groupes de classification de 13 catégories et moins. De plus, un temps de traitement moyen inférieur à 200 ms par image a été réalisé à partir de ces mêmes groupes de classification.

Le logiciel *Biotaxis* est proposé comme une nouvelle solution pour classifier rapidement des images de phytoplancton.

Mots-clés : Prétraitement, Discrimination, SVM, Précision, Temps de traitement, Phytoplancton

REMERCIEMENTS

La réalisation d'une recherche de maîtrise nécessite beaucoup de travail. C'est grâce à l'aide, au support et au temps donnés par certaines personnes, que cette recherche a pu être complétée. J'aimerais donc en profiter pour remercier chacune de ces personnes.

D'abord, je remercie M. Ruben Gonzalez-Rubio, mon directeur de recherche, qui a donné de son temps sans compter. Il m'a guidé tout au long de ma recherche et ramené à l'essentiel de la recherche lorsque je bifurquais. Merci également à M. Yannick Huot, membre du jury de cette recherche, qui a donné un thème aux images classifiées dans cette recherche et contribué à l'augmentation de mes connaissances sur les phytoplanctons. Je voudrais ensuite remercier M. Daniel Dalle, membre du jury de cette recherche, pour avoir aidé à améliorer la rédaction de ma définition du projet de recherche (DPR). Merci également à Mme Heidi Sosik, du Woods Hole Oceanographic Institution (WHOI) de Boston, pour avoir contribué à cette recherche en fournissant les groupes d'images d'entraînement et de test du classificateur.

Finalement, j'aimerais remercier la FQRNT (Fonds de recherche du Québec - Nature et technologies) pour m'avoir attribué une bourse de recherche. Ce support financier fut grandement apprécié.

TABLE DES MATIÈRES

1	INTRODUCTION	1
1.1	Mise en contexte et problématique	1
1.2	Définition du projet de recherche	2
1.3	Objectifs du projet de recherche	2
1.4	Objectifs spécifiques	3
1.5	Plan du mémoire de recherche	4
2	CONCEPTS GÉNÉRAUX DE LA CLASSIFICATION D'IMAGES DE PHYTOPLANKTON	5
3	ÉTAT DE L'ART	9
3.1	Prétraitement	11
3.1.1	Filtrage et détection de contour	12
3.1.2	Seuillage d'image et retrait des particules distantes	13
3.1.3	Tracé de contour de l'élément principal	15
3.1.4	Acquisition des caractéristiques de l'élément principal	16
3.2	Discrimination de catégories	19
3.2.1	But recherché par la discrimination de catégories	20
3.2.2	Calcul des limites d'une caractéristique	20
3.2.3	Traitement de discrimination	21
3.2.4	Arbre décisionnel à intervalles	21
3.3	Classification	23
3.3.1	Classification SVM linéaire	24
3.3.2	Classification SVM non linéaire	26
3.3.3	Traitement de classes multiples	28
3.4	Précision et temps de traitement	28
3.5	Résumé de l'état de l'art	30
4	DÉVELOPPEMENT LOGICIEL	32
4.1	Analyse préliminaire	32
4.1.1	Quelles images utiliser pour la classification ?	33
4.1.2	Identification des étapes de traitement requises par le logiciel	35
4.1.3	Choix des langages de programmation et du logiciel de développement	39
4.2	Conception du logiciel <i>Biotaxis</i>	41
4.2.1	Communication entre les langages Java et C++	41
4.2.2	Chargement d'images	43
4.2.3	Prétraitement	44
4.2.4	Discrimination	48
4.2.5	Classification SVM	50
4.2.6	Précision de la classification et temps de traitement	53

4.2.7	Description des options de traitement du logiciel <i>Biotaxis</i>	55
4.3	Tests des différentes phases de classification	63
4.4	Méthode de classification d'image avec le logiciel <i>Biotaxis</i>	64
4.5	Bilan du développement logiciel	66
5	RÉSULTATS ET ANALYSE	67
5.1	Présentation des résultats obtenus	67
5.1.1	Cadre expérimental	67
5.1.2	Spécification du système informatique	69
5.1.3	Solution 1 - Classification AVEC discrimination	69
5.1.4	Solution 2 - Classification SANS discrimination	70
5.2	Analyse des résultats	71
5.2.1	Comparaison des résultats des deux solutions	71
5.2.2	Interprétation des résultats	73
6	DISCUSSION	75
6.1	Les bonnes décisions	75
6.2	Les décisions à considérer	76
7	CONCLUSION	78
7.1	Récapitulatif	78
7.2	Contributions	79
7.3	Travaux futurs	80
A	DONNÉES	81
B	DONNÉES (Suite)	82
C	Analyse des métriques du logiciel <i>Biotaxis</i>	83
	LISTE DES RÉFÉRENCES	84

LISTE DES FIGURES

2.1	Schéma de la suite d'étapes de classification d'une image selon les solutions 1 et 2 proposées dans le cadre de cette recherche	6
2.2	Schéma simplifié du processus de classification d'une image	7
3.1	<i>Kernel</i> 3 x 3 du filtre Sobel	13
3.2	<i>Kernel</i> 3 x 3 du filtre Prewitt	13
3.3	Séquence de seuillage de la méthode hystérésis	14
3.4	Démonstration du calcul des Seuils S_b et S_h	15
3.5	Méthode <i>Convex Hull</i> appliquée à une image de phytoplancton	16
3.6	<i>Convex Hull</i> et tracé de contour	17
3.7	Trouver la plus longue diagonale du <i>Convex Hull</i>	17
3.8	Première caractérisation	17
3.9	Étapes A et B de création d'un arbre décisionnel à intervalles	22
3.10	Étape C de création d'un arbre décisionnel à intervalles	22
3.11	Classificateur SVM avec hyperplan optimal et marge maximale	25
3.12	Mappage d'un problème non linéaire vers espace supérieur	26
3.13	Matrice de confusion et calcul de la précision	29
4.1	Exemple d'images des 21 catégories de phytoplancton utilisées pour cette recherche	33
4.2	Organisation des fichiers images en ensemble d'images de 21 catégories . .	35
4.3	Séquence d'étapes d'entraînement d'un ensemble d'images menant à la création d'un groupe de classification	37
4.4	Séquence d'étapes de test d'un ensemble d'images et d'un groupe de classification menant à la classification d'images (comprend l'intégration de la séquence d'étapes selon les solutions 1 et 2).	39
4.5	Schéma de concept simplifié du logiciel <i>Biotaxis</i>	41
4.6	Interface après la sélection de l'onglet « <i>Analysis</i> » (logiciel <i>Biotaxis</i>) . . .	55
4.7	Interface après la sélection de l'onglet « <i>Training</i> » (logiciel <i>Biotaxis</i>) . . .	58
4.8	Interface après la sélection de l'onglet « <i>Testing</i> » (logiciel <i>Biotaxis</i>)	61
4.9	Exemple d'informations produites en console lors de la génération d'arbres décisionnels à intervalles (logiciel <i>Eclipse</i>)	64
5.1	Comparaison de la précision de la classification atteinte en fonction du nombre de catégories à classer (par solution et filtre)	72
5.2	Comparaison du nombre d'images traitées par seconde en fonction du nombre de catégories à classer (par solution et filtre)	72
C.1	Métriques du module en Java de <i>Biotaxis</i> (le 19 mai 2014)	83

LISTE DES TABLEAUX

2.1	Précision de la classification d'images de phytoplancton obtenue avec l'usage de la méthode SVM selon les articles à l'étude ayant obtenus plus de 70 % de précision de la classification	6
3.1	Compilation des phases de prétraitement de deux articles à l'étude	11
3.2	Phases de prétraitement et méthodes retenues pour cette recherche	12
3.3	Caractéristiques obtenues à l'aide du <i>Convex Hull</i> d'un objet	18
3.4	Discrimination de catégorie selon les limites d'une caractéristique	19
3.5	Quelques <i>kernels</i> fréquemment utilisés	27
5.1	Liste des catégories appartenant aux différents groupes de classification TEST	68
5.2	Résultats obtenus avec la solution 1 selon le filtre utilisé (précision et temps de traitement)	69
5.3	Résultats obtenus pour la solution 2 selon le filtre utilisé (précision et temps de traitement)	70
A.1	Resultats obtenus avec la solution 1 (AVEC discrimination)	81
B.1	Resultats obtenus avec la solution 2 (SANS discrimination)	82

LISTE DES SYMBOLES

Symbole	Définition
\mathbb{R}^n	L'univers de données possibles de n dimension(s)
S	Un échantillon de données de l'univers \mathbb{R}^n
D	Un groupe de formation préidentifié d'un classificateur SVM
f	Une fonction cible d'un classificateur SVM
\hat{f}	Modèle d'approximation de la fonction f
$*$	Exposant d'une variable lorsque sa valeur est optimale
\vec{x}	Représente un vecteur dans l'univers \mathbb{R}^n
\vec{w}	Vecteur normal de l'hyperplan optimal
b	Décalage entre l'hyperplan optimal et l'origine
$sgn()$	Donne le signe de l'élément dans la parenthèse (soit -1, 0, +1)
Φ	Mappage non linéaire
$\Phi(\vec{x})$	Application du mappage non linéaire sur le vecteur \vec{x}
α	Multiplicateur de Lagrange
c	Constante de coût du kernel polynomial non homogène
β	Largeur de bande pour le kernel gaussien
σ	Écart-type
\bar{x}	Moyenne

LISTE DES ACRONYMES

Acronyme	Définition
WHOI	Woods Hole Oceanographic Institution
IFCB	Imaging FlowCytobot
k-NN	k-nearest neighbor (k plus proches voisins)
SVM	Support Vector Machine (Machines à vecteurs de support)
CPU	Central Processing Unit
GPU	Graphics Processing Unit
TP	True Positive (Vrai Positif)
TN	True Negative (Vrai Négatif)
FP	False Positive (Faux Positif)
FN	False Negative (Faux Négatif)
JNI	Java Native Interface
JVM	Java Virtual Machine
ROI	Region of Interest
MVC	Modèle - Vue - Contrôleur
min	Minimum
max	Maximum
<i>png</i>	Portable Network Graphics
<i>csv</i>	Comma-Separated Values
<i>bio</i>	Format de fichier généré par <i>Biotaxis</i>

CHAPITRE 1

INTRODUCTION

1.1 Mise en contexte et problématique

L'étude de la biodiversité aquatique présente encore à ce jour, son lot de questionnements. Les percées technologiques des dernières années ont permis l'avènement d'une multitude d'appareils conçus dans le but d'aider les chercheurs à résoudre ces interrogations. Parmi ces appareils, ceux dédiés à la capture automatisée d'images marines ont contribué à l'émergence de nouvelles approches de classification. Dans le cadre de cette recherche, ce sont les approches liées à la classification d'images de phytoplancton qui ont servi de base au développement de ce projet.

Un phytoplancton est un micro-organisme du plancton végétal que l'on retrouve dans les différentes étendues d'eau de notre planète [Falkowski et Raven, 2007]. On dénombre des milliers d'espèces, mais leur variété sur un site donné est fonction de leur habitat (eau douce, salée, lac, rivière, etc.). Ils sont si petits qu'il est possible de dénombrer une centaine, voir des milliers de spécimens dans une seule gouttelette d'eau. Leur rôle est d'importance, car ils sont à la base de la chaîne alimentaire aquatique et parce qu'ils consomment plus de la moitié du CO_2 rejeté dans l'atmosphère. Une des questions à l'étude pour le phytoplancton est de connaître son évolution dans une étendue d'eau.

Il y a quelques années, la classification des images de phytoplancton était réalisée par un opérateur qui analysait et classait chacune de ces images. Mais la puissance toujours croissante de l'ordinateur combinée avec l'implémentation de modèles de classification a permis la création d'outils logiciels pouvant traiter automatiquement une grande quantité d'images. À partir de ces modèles, des groupes de chercheurs ont travaillé sur différentes approches afin d'avoir des outils leur permettant de faire une analyse qualitative, quantitative et temporelle du phytoplancton. Plusieurs approches ont été essayées et testées, mais il reste encore place à améliorations du point de vue de la précision de la classification et du temps de traitement.

1.2 Définition du projet de recherche

En partant de ces observations, une analyse de la précision de la classification des différentes approches à l'étude a permis de déterminer que dans leur ensemble, les processus employés offrent la possibilité d'obtenir une précision de la classification de 87 %. L'analyse de ces études sera précisée au chapitre 2.

Ce nombre représentant la précision de la classification est très important, car plus il est grand, plus la classification des phytoplanctons par catégorie sera fiable. Ainsi, la projection sur un graphique de l'évolution du nombre de phytoplanctons par catégorie en fonction du temps donnera une courbe beaucoup plus représentative de la variation de la population. Permettant une meilleure compréhension de l'interaction entre ces micro-organismes et leur environnement. Cette projection en temps réel de l'évolution de la population sous forme graphique serait en soi un outil très utile pour la communauté scientifique travaillant sur le sujet.

Avant de pouvoir faire cette projection en temps réel, il est impératif de connaître le temps de traitement requis par le processus de classification. Les articles à l'étude traitant de la classification de phytoplancton n'ont pas permis d'identifier les temps de traitement requis. Une étude de ce temps et de la précision de la classification lors de l'usage de processus donnés permettrait de faire un choix de méthodes de traitement plus éclairé lorsque le niveau d'importance de l'un ou l'autre de ces deux aspects varie. C'est à partir de cette observation que la question de recherche suivante fut établie :

« Comment améliorer ou conserver la précision de classification actuelle des images de phytoplancton, tout en diminuant le temps de traitement requis ? »

1.3 Objectifs du projet de recherche

L'objectif global est donc d'étudier le temps de traitement requis, dépendamment des processus de classification utilisés, afin de présenter une base décisionnelle quant au bien-fondé de l'usage ou non de ces processus.

Pour atteindre cet objectif, le projet réalisé a d'abord nécessité l'identification de balises, qui ont ensuite été utilisées pour guider l'ensemble du processus de classification par catégorie des images de phytoplancton. Les différentes balises représentant les objectifs secondaires du projet sont :

- d'atteindre une précision de la classification d'environ 87 % ou plus,

- de viser un temps de traitement moyen de 200 ms ou moins par image,
- de concevoir un logiciel de classification d'images.

Afin d'atteindre l'objectif principal et de respecter les objectifs secondaires, la recherche fut orientée de la façon suivante. D'abord, il convenait d'avoir un outil logiciel qui permettrait de comptabiliser les résultats obtenus et qui offrirait l'assurance du respect des objectifs secondaires. Le logiciel *Biotaxis*¹ a donc été conçu dans cette optique. Soit d'établir la précision de la classification d'images de phytoplancton selon différents processus employés, tout en générant des informations sur le temps de traitement requis pour chacun de ces processus.

Suite à cette approche, il convenait de déterminer les processus à utiliser pour faire la classification d'images et ensuite, de faire l'analyse des résultats obtenus. C'est en se fondant sur ces prémisses que le projet de recherche fut réalisé.

1.4 Objectifs spécifiques

Cette recherche proposée vise à la réalisation du logiciel *Biotaxis*. Ce logiciel offre la possibilité à l'utilisateur de choisir l'une parmi deux solutions proposées pour effectuer la classification d'images de phytoplancton. Les phases de traitements appliqués étant :

- Solution 1 : comprenant les phases de prétraitement, de discrimination de catégories et de classification.
- Solution 2 : comprenant les phases de prétraitement et de classification.

Au terme de la classification d'images, il est possible de compiler les résultats de la précision de la classification et de réaliser une comparaison avec ceux générés par l'usage d'autres filtres, groupes d'images ou solution de traitement. Cette comparaison permet d'identifier la meilleure combinaison (images, filtres, solution) pour atteindre la précision de la classification ciblée.

En plus du logiciel *Biotaxis*, une autre contribution originale est la présentation d'une analyse comparative de la précision de la classification et du temps de traitement selon le nombre de catégories utilisé par groupe. Cette analyse permet d'établir la relation existant entre :

¹Afin d'éviter toute ambiguïté entre la désignation des projets référés avec notre projet, le nom *Biotaxis* sera utilisé. Ce nom réfère à la définition suivante : « la classification d'organisme vivant en fonction de leur caractéristique physique » [Encyclopedia, 2011].

- La précision de la classification versus le nombre de catégories à classifier.
- Le nombre d'images traitées par seconde versus le nombre de catégories à classifier.

1.5 Plan du mémoire de recherche

Ce mémoire de recherche a été organisé de la façon suivante. On retrouve tout d'abord le chapitre du « Concept généraux de la classification d'images de phytoplancton », qui est une introduction aux différents concepts ayant été utilisés dans ce projet. Ensuite, on passe au chapitre de l'« État de l'art », qui est une synthèse des avancées recensées dans la littérature qui ont été utilisées dans ce projet. Par la suite vient le chapitre sur le « Développement logiciel » qui présente l'application des concepts théoriques de l'état de l'art ayant mené à la réalisation du logiciel *Biotaxis*. Le chapitre suivant est « Résultats et Analyse » où sont présentés les résultats obtenus dans cette recherche et l'interprétation de ceux-ci. Puis vient ensuite le chapitre de « Discussion » qui est une évaluation de certaines décisions prises en cours de projet. Finalement, le mémoire se termine par le chapitre de « Conclusion ».

CHAPITRE 2

CONCEPTS GÉNÉRAUX DE LA CLASSIFICATION D'IMAGES DE PHYTOPLANKTON

Ce chapitre présente certains concepts nécessaires à la compréhension du sujet de recherche. Ce survol des idées maîtresses du projet de recherche, ainsi que leur interaction a donc pour but de permettre au lecteur d'avoir une idée globale du projet et de faciliter la lecture du chapitre suivant, traitant de l'état de l'art.

Ainsi, l'élément principal du projet *Biotaxis* est la classification d'images d'organismes vivants, soit les phytoplanctons. À ce jour, plusieurs méthodes sont disponibles pour classer des images. Parmi celles-ci on retrouve la méthode des k plus proches voisins ($k - NN$) [Cover et Hart, 1967], les réseaux de neurones [Rosenblatt, 1958] et les machines à vecteurs de support (SVM) [Vapnik, 1995], qui a été la méthode sélectionnée. Ce choix se fonde sur deux raisons majeures :

- Tous les articles de classification d'images de phytoplancton à l'étude pour cette recherche ont utilisé cette méthode.
- La qualité de la précision obtenue lors de son usage pour la classification d'images de phytoplancton (voir le tableau 2.1).

Pour illustrer cette qualité de la précision, voici un tableau énumérant les résultats qui ont été publiés dans des articles à l'étude. Les articles présentés ont été sélectionnés en fonction des détails donnés sur le nombre d'échantillons et de catégories utilisés, ainsi qu'au fait d'avoir obtenu une précision de la classification supérieure à 70 %.

Comme on peut le constater, les pourcentages de précision cités dans le tableau 2.1 sont très élevés. Il y a cependant certaines divergences dans les approches employées pour obtenir ces résultats. Voici une liste des divergences prédominantes ressorties lors de l'étude des articles du tableau 2.1 :

- Le type et le nombre de catégories taxonomiques¹ utilisées.
- Le nombre d'échantillons employés pour entraîner le classificateur.

¹Relatif à la taxonomie, science de la classification des êtres vivants.

Article	Nb. Échantillons	Nb. Catégories	Précision
Alvarez <i>et al.</i> [2012]	526	6	86 %
Su <i>et al.</i> [2010]	1629	4	89 %
Kang <i>et al.</i> [2010]	1300	35	86.59 %
Sosik et Olson [2007]	6600	22	88 %
Hu et Davis [2005]	20000	7	72 %
Luo et Kramer [2004]	6000	5	75.57 %

Tableau 2.1 Précision de la classification d'images de phytoplancton obtenue avec l'usage de la méthode SVM selon les articles à l'étude ayant obtenus plus de 70 % de précision de la classification

- Le choix des méthodes utilisées pour le prétraitement des images.
- Le type et la quantité d'attributs de caractérisation des images employés.

À défaut de connaître l'impact des divergences énumérées ci-haut, une analyse des approches de classification citées dans les articles listés dans le tableau 2.1 a été réalisée pour cette recherche et synthétisée par le schéma de la figure 2.1. Ce schéma présente les deux solutions de classification proposées, ainsi que les séquences d'opérations propres à chacune de ces solutions.

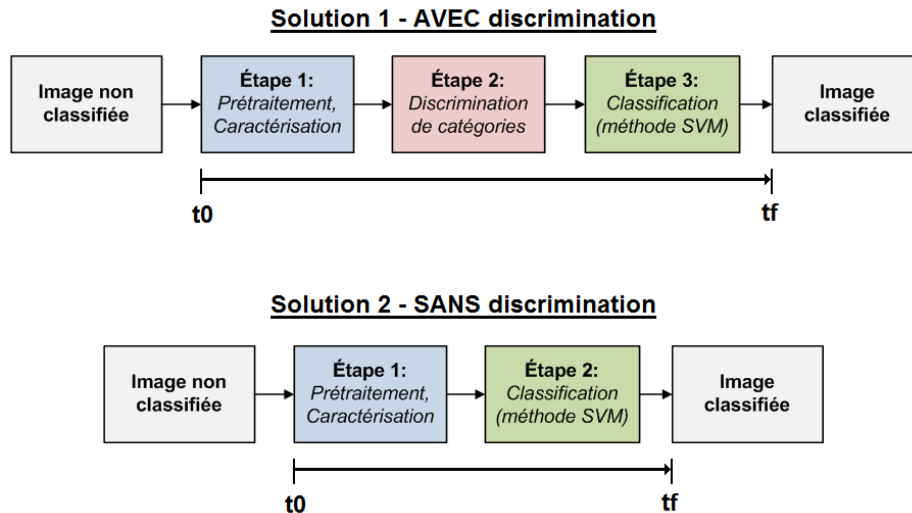


Figure 2.1 Schéma de la suite d'étapes de classification d'une image selon les solutions 1 et 2 proposées dans le cadre de cette recherche

Les trois étapes de classification citées dans la figure 2.1 se résument comme suit :

- Le prétraitement comprend plusieurs phases de traitement tel que : la réduction du bruit, l'extraction de la zone d'intérêt d'une image (conserver uniquement le phytoplancton) et la caractérisation de la zone d'intérêt.

- La discrimination de catégories ayant comme objectif : la réduction du nombre de catégories d'appartenance d'une image en fonction d'attributs spécifiques (tel que la grandeur et la forme de la zone d'intérêt). Le but étant de discriminer des catégories afin que la classification SVM se fasse uniquement sur les catégories restantes. Il est espéré que cette étape permettra de diminuer le temps de traitement.
- La classification attribue une catégorie à l'image traitée en fonction du résultat obtenu par la méthode SVM.

La figure 2.2 présente un exemple simplifié d'un classificateur d'images contenant une forme rectangulaire. Ce schéma donne une idée générale du fonctionnement d'un classificateur. Les traitements énoncés dans chacune des étapes du schéma sont ceux employés par les classificateurs utilisés dans cette recherche (solution 1 ou 2).

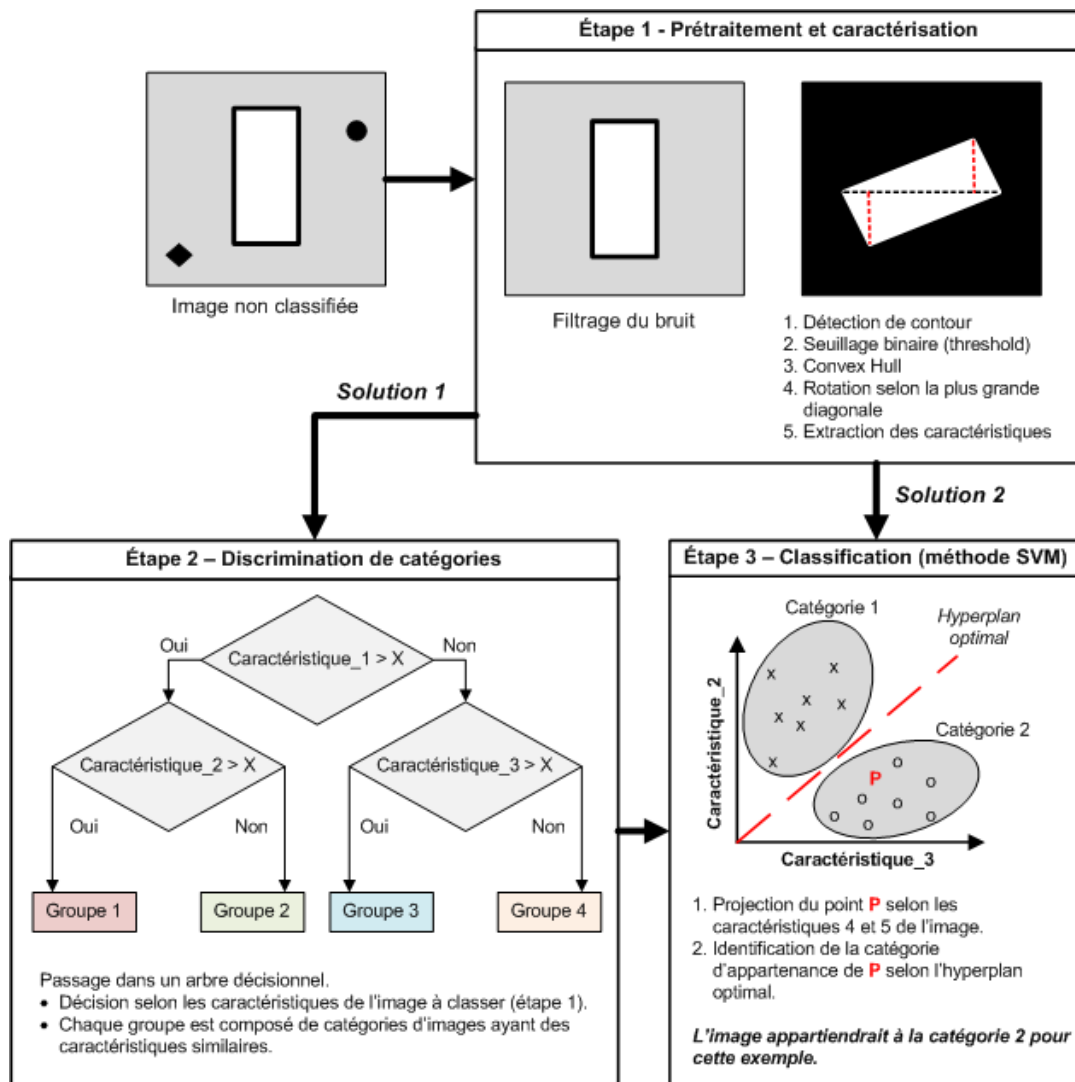


Figure 2.2 Schéma simplifié du processus de classification d'une image

Si l'on se réfère au schéma de la figure 2.1, on y retrouve une échelle de temps comprenant deux jalons pour marquer la séquence temporelle à l'étude. Puisque l'un des objectifs du projet *Biotaxis* est d'obtenir un temps de traitement moyen de la classification de 200 ms. Alors le temps compris entre t_0 et t_f , se doit d'être quantifié pour être en mesure de déterminer le temps de traitement moyen par image, par catégorie d'images et par groupe de classification (qui un ensemble de catégories).

En conclusion, ce chapitre avait pour but de présenter les concepts globaux de la classification d'images de phytoplancton. Le choix de la méthode de classification utilisée a été abordé (méthode SVM), ainsi qu'une présentation des résultats de la précision de la classification obtenus pour certains articles (ayant utilisé la méthode SVM). Ensuite, deux solutions de classification ont été proposées suivies par une description des différentes étapes de traitement requises par chacune d'elle. Ce chapitre avait donc pour objectif de faciliter la compréhension des concepts généraux de la classification, avant de passer au chapitre suivant (l'« État de l'art ») qui traite de façon détaillée, l'ensemble des concepts qui ont été abordés.

CHAPITRE 3

ÉTAT DE L'ART

Chaque jour, une grande quantité d'images sont générées quotidiennement par des appareils automatisés de capture d'images de phytoplancton. Il suffit de citer comme exemple le IFCB [Olson et Sosik, 2007], qui est un appareil pouvant capturer 10 000 images à l'heure (soit environ 240 000 images par jour), pour comprendre que la classification manuelle de l'ensemble de celles-ci représente une tâche presque impossible à réaliser, à défaut d'avoir un grand nombre de ressources humaines attitrées à cette tâche.

Puisque cette énorme quantité d'images reste inutilisable si aucune caractérisation et classification de celles-ci n'est faite, des groupes de chercheurs ont donc travaillé sur des solutions possibles afin d'automatiser ce processus. L'analyse de quelques-unes des solutions avancées illustre la possibilité de combiner certaines des méthodes et approches proposées en vue de réaliser l'implémentation d'un logiciel de classification.

Cependant, avant de pouvoir réaliser un tel outil, il est important de cibler certains critères spécifiques permettant d'évaluer le logiciel. Parmi les critères possibles, deux d'entre-eux requièrent une attention particulière :

- Le premier est la précision de la classification, qui va de pair avec la qualité de l'analyse, car plus il y a d'images « bien » classées, plus l'évaluation de la variation de populations dans le temps est précise.
- Le deuxième est le temps de traitement. Ce paramètre détermine en combien de temps la classification d'une image est effectuée. Par exemple, si le logiciel peut classer 1 000 images à l'heure, mais que 2 000 images sont générées au cours de cette même période de temps, alors seulement une fraction d'entre elles pourra être traitée à l'heure (50 % pour cet exemple).

Ces deux critères sont intimement liés, car l'usage d'une méthode ayant un impact sur la précision aura incidemment un impact sur le temps de traitement. Il convient alors de bien choisir les méthodes et approches utilisées en vue d'accroître la précision et de minimiser le temps d'opération. Certaines décisions pourraient ainsi mener à une légère perte de précision en vue d'obtenir un large gain en temps de traitement et la situation inverse est également possible.

Pour mener à bien ce projet, les deux critères cités ci-haut ont été pris en compte. La revue de littérature a été orientée vers les ouvrages traitant de méthodes de classification des phytoplanctons, utilisant les machines à vecteurs de support (SVM) comme modèle d'apprentissage supervisé. Parmi les ouvrages étudiés, seulement ceux ayant atteint 70 % et plus de précision de la classification ont été retenus comme référence pour cette recherche. Chacun d'eux a appliqué leurs traitements sur des images bidimensionnelles et a utilisé des métriques de discrimination taxonomique pour favoriser l'augmentation de la précision de leur processus de classification.

Les prochaines sections de ce chapitre traitent des différentes étapes du processus de classification (tel que défini à la figure 2.1 du chapitre 2). Les deux solutions considérées dans cette recherche et détaillées au chapitre 2, comportent la série d'étapes suivantes :

- Solution 1 (avec discrimination) :

1. Prétraitement
2. Discrimination
3. Classification

- Solution 2 (sans discrimination) :

1. Prétraitement
2. Classification

En se basant sur les étapes de la solution 1, cela permet de couvrir également les étapes de la solution 2. C'est pourquoi les sections suivantes de l'état de l'art présentent l'ensemble des étapes utilisées par la solution 1. Évidemment, peu importe la solution choisie, le but commun recherché est de prendre une image et de la manipuler en vue de déterminer sa catégorie d'appartenance.

La présentation des sections suivantes suit donc l'ordre d'opération de la solution 1. Les méthodes citées dans chacune de ces sections font référence à des ouvrages sur la classification de phytoplanctons, ainsi qu'à d'autres ouvrages ciblant un type de traitement bien spécifique. En résumé, le choix des méthodes, citées dans l'état de l'art, résulte de l'analyse et la comparaison de chacun des traitements utilisés dans les ouvrages sur le classement de phytoplanctons, avec des ouvrages dédiés spécifiquement au traitement comparé. La sélection finale du procédé ou de la méthode pour chaque traitement fut basée sur leur contribution à l'augmentation de la précision de la classification et/ou leur capacité à diminuer le temps de traitement.

3.1 Prétraitement

Pour bien définir les différents processus caractérisant cette étape, une analyse des articles à l'étude sur le sujet a été menée afin de cibler, puis de déterminer la suite logique des traitements nécessaires. Le passage par ces processus vise l'obtention de valeurs numériques permettant de caractériser les différents paramètres d'une image donnée. Le tableau 3.1 présente la séquence d'opérations de prétraitement citée dans deux des articles à l'étude ayant décrit l'ensemble de leurs procédés (certaines parties présentes également une référence).

Phases de prétraitement	Articles de référence	
	Sosik et Olson [2007]	Su <i>et al.</i> [2010]
1. Filtrage et détection de contour	« Noisecompensated phase congruency » [Kovesi, 1999]	Méthode Canny de détection de contour [Canny, 1986]
2. Seuillage de l'image	Seuillage (« thresholding »)	
3. Retraits des particules distantes	Opérateurs morphologiques (fermeture, dilatation et amincissement)	Opérateurs morphologiques (dilatation, fermeture) et détection de la région d'intérêt (<i>ROI</i>)
4. Tracé de contour de l'élément principal	Tracé des contours simplifiés de l'élément principal en fonction du premier 10 % du descripteur de Fourier [Gonzalez <i>et al.</i> , 2009]	
5. Acquisition des caractéristiques de l'élément principal	Caractérisation de l'élément principal (périmètre, aire, longueur, etc.)	

Tableau 3.1 Compilation des phases de prétraitement de deux articles à l'étude

Note : Le choix des deux articles utilisés pour la création du tableau 3.1 est basé sur la description des méthodes employées lors de la phase de prétraitement. Tel que défini dans ce même tableau, l'article de Sosik et Olson [2007] est le plus exhaustif sur les méthodes utilisées lors des différentes phases de prétraitement.

Suite à l'analyse des différentes phases de prétraitement énumérées dans le tableau 3.1, l'étude subséquente fut de comparer dans la littérature les méthodes citées dans ce tableau versus d'autres méthodes au rendu similaire en fonction du temps de traitement requis. Cette étude sur le prétraitement des images a permis d'identifier et choisir les méthodes utilisées dans cette recherche. Le tableau 3.2 fait état de ces choix de méthodes selon les différentes phases de prétraitement.

Phases de prétraitement	Méthodes retenues et références
1. Filtrage et détection de contour	Prewitt et Sobel [Prewitt, 1970; Sobel et Feldman, 1968]
2. Seuillage de l'image et retraits des particules distantes	Seuillage par hystérésis [Pridmore, 2002]
3. Tracé de contour de l'élément principal	Méthode <i>Convex Hull</i> [de Berg <i>et al.</i> , 2008]
4. Acquisition des caractéristiques de l'élément principal	Caractérisation de l'élément principal selon le <i>Convex Hull</i> (aire, périmètre, etc.) [Luo, 1998]

Tableau 3.2 Phases de prétraitement et méthodes retenues pour cette recherche

Les sous-sections suivantes de l'état de l'art présentent chacune des phases de prétraitement du tableau 3.2 ci-haut mentionnées, en respectant la séquence des traitements énumérée.

3.1.1 Filtrage et détection de contour

Cette étape vise d'abord à filtrer une image en vue d'enlever le bruit, puis de faire la détection des contours de l'élément principal de l'image (soit le phytoplancton). L'article de Sosik et Olson [2007] cite l'usage de la méthode « noise-compensated phase congruency » pour réaliser les deux processus cités précédemment. Cependant, il est fait mention que cette méthode est très intensive en calcul de processeur, c'est pourquoi cette approche a été rejetée pour cette recherche. L'article de Su *et al.* [2010] relate l'usage de la méthode Canny pour faire le filtrage et la détection des contours de l'élément principal de l'image. Cependant, l'étude de l'usage de cette méthode faite par Sosik et Olson [2007] pour le traitement d'image de phytoplancton a démontré que la méthode Canny ne pouvait pas être optimisée pour éliminer le bruit et les variations d'intensité lumineuse présente autour de l'élément principal de l'image.

L'étude des autres pistes de solution permettant la détection de contour a permis d'identifier le filtre Sobel comme étant une méthode fréquemment utilisée pour ce type de tâche. De plus, ce filtre fait également partie de la suite de traitements utilisée par la méthode Canny [Canny, 1986]. Ces observations ont donc favorisé l'adoption de cette approche pour cette étape de prétraitement.

Le filtre Sobel est un masque de convolution que l'on applique à une image par un balayage en X et Y , afin de calculer le gradient d'intensité lumineuse de chaque point. L'image résultante fait ressortir les changements abrupts de luminosité dans l'image originale, donc les

contours probables d'une ou plusieurs formes. La figure 3.1 présente les *kernels* définissant le filtre Sobel ainsi que l'opérateur mathématique permettant d'obtenir le gradient normalisé pour chaque point d'une image traité à l'aide de cet opérateur [Sobel et Feldman, 1968].

$$\begin{array}{c}
 \mathbf{G}_x = \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 2 & 0 & -2 \\ \hline 1 & 0 & -1 \\ \hline \end{array}
 \end{array}
 \quad
 \begin{array}{c}
 \mathbf{G}_y = \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline \end{array}
 \end{array}
 \quad
 \mathbf{G} = \sqrt{G_x^2 + G_y^2} \text{ (Gradient normalisé)}$$

Figure 3.1 *Kernel 3 x 3* du filtre Sobel

À titre comparatif au filtre Sobel, le filtre Prewitt [Prewitt, 1970] a également été étudié dans cette recherche due aux similitudes des *kernels* 3x3 utilisés par ce dernier versus ceux du filtre Sobel (voir figures 3.1 et 3.2).

$$\begin{array}{c}
 \mathbf{G}_x = \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline -1 & 0 & 1 \\ \hline \end{array}
 \end{array}
 \quad
 \begin{array}{c}
 \mathbf{G}_y = \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -1 & -1 \\ \hline \end{array}
 \end{array}
 \quad
 \mathbf{G} = \sqrt{G_x^2 + G_y^2} \text{ (Gradient normalisé)}$$

Figure 3.2 *Kernel 3 x 3* du filtre Prewitt

3.1.2 Seuillage d'image et retrait des particules distantes

Le seuillage d'une image en niveau de gris consiste à remplacer chacun de ses pixels par la valeur 0 (noir) ou 255 (blanc) en fonction d'un seuil déterminé. Cette méthode a été utilisée par Sosik et Olson [2007] pour obtenir une image où la surface du phytoplancton est en blanc et l'arrière-plan en noir. Le seuillage d'une image est défini selon l'opérateur mathématique suivant :

Si $g(x, y)$ est la version de l'image après le seuillage de $f(x, y)$ selon un seuil T alors,

$$g(x, y) = \begin{cases} 255 & \text{si } f(x, y) \geq T \\ 0 & \text{sinon} \end{cases} \quad (3.1)$$

La qualité de l'extraction de la forme d'un phytoplancton dans une image est fonction du seuil établi. Pour déterminer la valeur de ce seuil, il faut se référer aux étapes de traitement de la méthode de seuillage choisi. Parmi les méthodes de seuillage possible, la méthode hystérésis a été utilisée pour cette recherche en raison de la qualité du rendu

après seuillage [Pridmore, 2002], à son usage par la méthode Canny [Canny, 1986] et à sa capacité d'éliminer les particules distantes (tel que des poussières) présentent dans l'image .

La méthode de seuillage hystérésis a la particularité d'utiliser 2 seuils au lieu d'un seul, soit un seuil bas (S_b) et un seuil haut (S_h). Dans le cas du traitement d'image de phytoplancton, le seuil du bas délimite le niveau de gris à partir duquel il y a une grande probabilité que les niveaux de gris inférieurs appartiennent au phytoplancton que l'on désire détecter. Puis, le seuil du haut délimite le niveau de gris à partir duquel les niveaux de gris inférieur ont une forte probabilité d'appartenir au phytoplancton. Ces deux seuils sont donc utilisés séparément sur l'image à traiter afin d'obtenir une première image basée sur le seuil S_b et une seconde basée sur le seuil S_h . Ensuite, tous les pixels blancs de l'image de seuillage S_h qui touchent à un pixel blanc de l'image S_b sont ajoutés à l'image de seuillage S_b pour obtenir une image résultante d'hystérésis. La figure 3.3 présente les résultats obtenus après chaque séquence de seuillage.

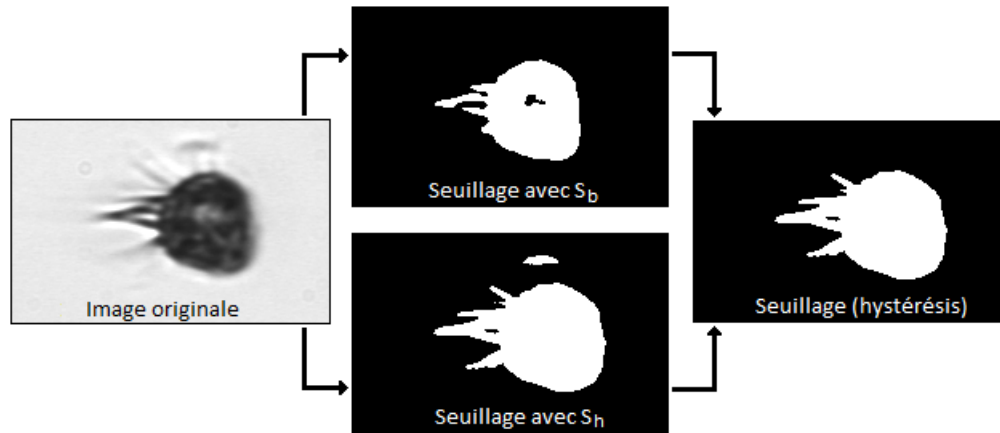


Figure 3.3 Séquence de seuillage de la méthode hystérésis¹

Pour calculer les seuils S_b et S_h , la méthode employée fut inspirée de l'étude de Pridmore [2002]. Voici les étapes du processus (voir la figure 3.4 comme référence) :

1. Tracer l'histogramme du niveau de gris de l'image à traiter.
2. Déterminer les niveaux de gris limite de l'histogramme, soit : le pixel ayant un niveau de gris le plus près du noir (X_i), le pixel ayant un niveau de gris le plus près du blanc (X_f).

¹Images obtenues à l'aide du logiciel Biotaxis (Ciliate, réf. 2006_052_145231_2649.png). À noter que l'image de seuillage S_h présente un groupe de points blancs au-dessus de la forme du phytoplancton et que ce groupe n'est pas présent dans l'image d'hystérésis, car aucun des points du groupe ne touche un point blanc de l'image de seuillage S_b .

3. Diviser en deux sections (bas et haut), selon le centre de X_f et X_i . Le centre étant déterminé selon l'équation 3.2, les limites délimitant l'intervalle du bas sont donc X_i et X_c et pour le haut X_c et X_f .

$$X_c = ((X_f - X_i)/2) + X_i \quad (3.2)$$

4. Calculer la moyenne \bar{x} et l'écart type σ de la section du bas et du haut. Pour la section du bas on calcule \bar{x}_b et σ_b et \bar{x}_h et σ_h pour le haut.
5. Déterminer les valeurs de seuil bas (S_b) et de seuil haut (S_h), selon les équations :

$$S_b = \bar{x}_b + 2\sigma_b \quad (3.3)$$

$$S_h = \bar{x}_h - 2\sigma_h \quad (3.4)$$

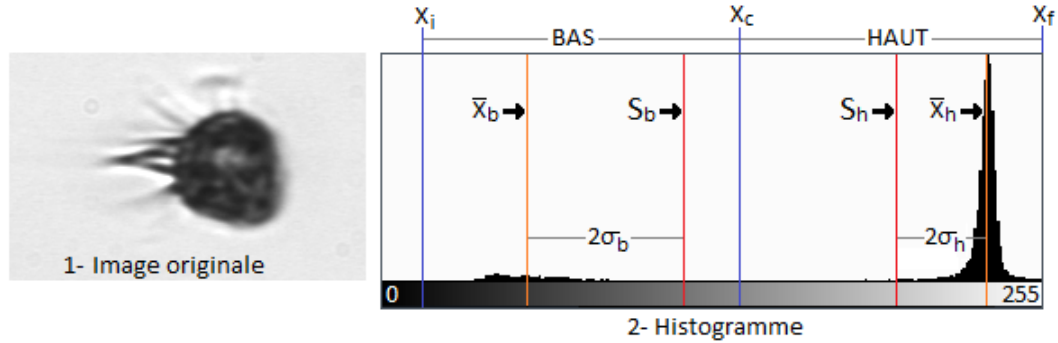


Figure 3.4 Démonstration du calcul des Seuils S_b et S_h ²

Pour ce qui est des particules distantes, puisque leur niveau de gris est presque toujours supérieur au niveau de gris de l'élément principal (dans le contexte des images utilisées dans cette recherche), alors le seuillage de l'image est suffisant pour les retirer.

3.1.3 Tracé de contour de l'élément principal

Une fois les étapes de filtrage et seuillage de l'image complétées, le traitement suivant doit permettre de délimiter le contour de l'élément principal, donc de définir l'espace minimal requis permettant d'englober cet élément. Sosik et Olson [2007] ont utilisé le descripteur de Fourier pour tracer ce contour, mais ce descripteur nécessite beaucoup de calculs. L'étape suivante consiste donc à déterminer quelle méthode permettrait de tracer un contour plus rapidement que la méthode du descripteur de Fourier. Les études dans la littérature sur le

²Images obtenues à l'aide du logiciel Biotaxis (Ciliate, réf. 2006_052_145231_2649.png)

sujet ont permis d'identifier la méthode *Convex Hull* comme étant une solution simple et rapide pour obtenir une enveloppe de contour d'objet principal. L'exemple de la figure 3.5 présente le résultat obtenu lors de l'usage de ce processus à une image de phytoplancton ayant passé précédemment par une étape de seuillage.



Figure 3.5 Méthode *Convex Hull* appliquée à une image de phytoplancton³

À la figure 3.5, l'image 3 présente la surface de couverture du phytoplancton en blanc sur fond noir avec un tracé de contour représentant le résultat de l'application de la méthode *Convex Hull* sur l'image 2. L'ouvrage de de Berg *et al.* [2008], présente une méthode pour effectuer le tracé de contour d'un objet (voir la figure 3.6). Le principe consiste à trouver d'abord un point extrême *A*, soit par exemple le pixel le plus haut de l'objet, puis d'obtenir la coordonnée du point *B* positionné au-dessus du point *A*, soit à la limite supérieure de l'image (bordure). Ensuite, il faut tracer un vecteur entre les points *A* et *B*, en partant du point *A*. Si aucune bordure de l'objet n'a été touchée lors du tracé du vecteur, alors le vecteur est effacé, puis le point *B* est déplacé de un pixel dans le sens horaire. Un nouveau vecteur est alors tracé et effacé entre les points *A* et *B* tant que le vecteur ne touchera pas à une bordure de l'objet. Lorsqu'il y a un contact entre le vecteur et l'objet, le tracé entre le point *A* et le point de contact est conservé, puis la coordonnée du point de contact devient la nouvelle coordonnée du point *A*. La séquence de tracé du vecteur entre les points *A* et *B* se poursuit jusqu'à ce que le point *B* ait complété une rotation complète de la bordure de l'image (retour au point de départ). L'image 2 de la figure 3.6 présente le concept d'un déplacement de vecteur pour obtenir le tracé de contour d'un objet.

3.1.4 Acquisition des caractéristiques de l'élément principal

L'étape faisant suite à la méthode *Convex Hull* est l'acquisition des caractéristiques de la zone d'intérêt délimitée par le tracé de contour. L'ouvrage de Luo [1998] définit très bien les étapes et formules permettant de caractériser un objet à partir de la méthode *Convex Hull*. Cette séquence d'étapes se présente comme suit :

³Images obtenues à l'aide du logiciel Biotaxis (Ciliate, réf. 2006_052_145231_2649.png)

⁴ Référence : [de Berg *et al.*, 2008]

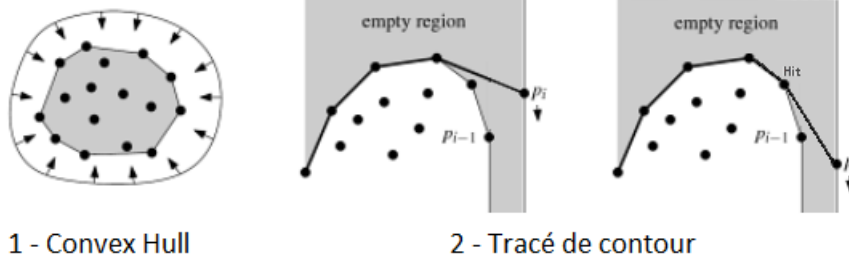


Figure 3.6 *Convex Hull* et tracé de contour⁴

1. Trouver la plus longue diagonale (D_{max}) du *Convex Hull* qui est obtenu par le calcul de la distance entre tous les coins du tracé du *Convex Hull*.

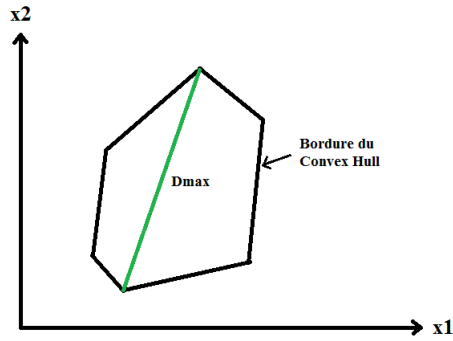


Figure 3.7 Trouver la plus longue diagonale du *Convex Hull*

2. Appliquer une rotation sur l'objet afin que la plus longue diagonale soit sur un plan horizontal. Puis trouver les valeurs verticales supérieures (V_t) et inférieures (V_b), calculer les largeurs maximales 1 et 2 (W_{max1} et W_{max2}) représentant les distances entre le plan horizontal et les valeurs V_t et V_b , puis calculer les distances entre l'origine et les largeurs maximales (W_{max1} et W_{max2}).

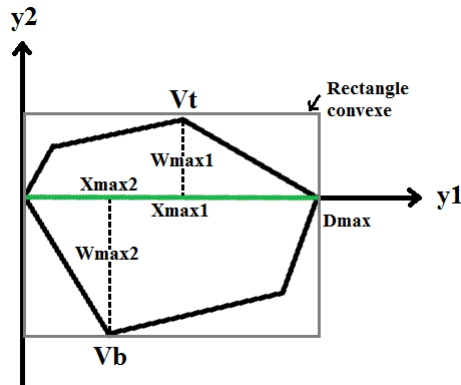


Figure 3.8 Première caractérisation

3. Ensuite, à l'aide des valeurs obtenues à l'étape 2, il est possible de calculer une série de caractéristiques propre à la zone d'intérêt. Le tableau 3.3 présente des exemples de caractéristiques pouvant être obtenues grâce à l'usage de la méthode *Convex Hull*.

No.	Caractéristique	Définition	Formule
1	D_{max}	Longueur de la diagonale maximale (en pixels)	
2	W_{max1}	Largeur maximale	$W_{max1} = X_{max1}/D_{max}$
3	W_{max2}		$W_{max2} = X_{max2}/D_{max}$
4	W_s	Largeur de l'objet	$W_s = W_{max1} + W_{max2}$
5	S_h	Symétrie horizontale	$S_h = W_{max2}/W_{max1}$
6	S_{v1}	Demi-Symétrie verticale	$S_{v1} = X_{max1} - 1/2$
7	S_{v2}		$S_{v2} = X_{max2} - 1/2$
8	A_{c1}	Aire du <i>Convex Hull</i> (partie minimale et maximale)	$A_{c1} = \int_0^1 g_1(y_1)dy_1$
9	A_{c2}		$A_{c2} = \int_{-1}^0 g_2(y_2)dy_2$
10	A_c	Aire totale du <i>Convex Hull</i>	$A_c = A_{c1} + A_{c2}$
11	A_p	Aire de l'objet (soit la forme délimitée par le tracé de contour)	
12	P_c	Périmètre du <i>Convex Hull</i>	Somme de la longueur des vecteurs de contour.
13	R_{cr1}	Ratio du <i>Convex Hull</i> versus son rectangle convexe	$R_{cr1} = \frac{A_{c1} - (D_{max} * W_{max1})/2}{(D_{max} * W_{max1})/2}$
14	R_{cr2}		$R_{cr2} = \frac{A_{c1} - (D_{max} * W_{max2})/2}{(D_{max} * W_{max2})/2}$
15	CDG_a	Déficiance globale du <i>Convex Hull</i>	$CDG_a = (A_c - A_p)/A_c$
16	R_{apc}	Facteur de l'aire de la forme versus le périmètre du Convex Hull	$R_{apc} = A_c/P_c^2$
17	C_{mean}	Moyenne des courbes du <i>Convex Hull</i>	$C_{mean} = \frac{1}{N} \sum_{i=1}^N Cur(v_i)$
18	F_{dim1}	Dimension fractale (selon demi-périmètre haut et bas)	$F_{dim1} = \log(peri_1)/\log(D_{max})$
19	F_{dim2}		$F_{dim2} = \log(peri_2)/\log(D_{max})$
20	Corner	Somme des coins du <i>Convex Hull</i> (max : 360 pour un cercle)	

Tableau 3.3 Caractéristiques obtenues à l'aide du *Convex Hull* d'un objet⁵

Note : les caractéristiques obtenues sont propres à l'objet d'intérêt, mais la représentativité de ces valeurs est fonction de l'image obtenue après les étapes de filtrage et de seuillage. Une image résultante n'ayant pas une représentation fidèle de la surface de l'objet d'intérêt

⁵Référence [Luo, 1998]

dû à une déformation ou la présence de particules causant un mauvais tracé du *Convex Hull*, occasionnera l'acquisition de valeurs de caractérisation erronées.

3.2 Discrimination de catégories

Suite au prétraitement, certaines approches étudiées passent par une étape de discrimination des catégories en fonction d'attributs spécifiques (caractéristiques) [Alvarez *et al.*, 2012; Hu et Davis, 2005; Sosik et Olson, 2007]. Certains critères sont utilisés comme discriminant par plusieurs chercheurs et d'autres non. Pour cette recherche, ce sont les caractéristiques énumérées dans le tableau 3.3 qui sont utilisées comme discriminants.

En analysant le tableau 3.3, on constate qu'il y a plusieurs discriminants possible. Chacun d'eux peut-être employé à la création d'un arbre décisionnel servant à départager les différentes catégories. En fonction des caractéristiques d'une image traitée, certaines catégories seront éliminées et d'autres conservées selon les résultats de comparaison. Par élimination, on désigne le retrait d'une catégorie comme candidat d'appartenance possible pour une image traitée, lorsque la valeur de l'une des caractéristiques de l'image traitée se situe à l'extérieur des limites minimale et maximale de cette même caractéristique définie pour une catégorie donnée (le calcul des limites est abordé à la sous-section 3.2.2) . Voici un exemple simplifié de cette logique :

Par exemple, si la caractéristique D_{max} (diagonale maximale) d'une image traitée mesure 250 pixels. Alors si l'on remplace la variable X dans le tableau 3.4 par la valeur de 250, la logique décisionnelle sera :

Catégorie	Limites pour D_{max}	Décision
Catégorie 1	$100 \leq X \leq 230$	+0 pour cette catégorie (D_{max} est trop grand)
Catégorie 2	$180 \leq X \leq 260$	+1 pour cette catégorie (limites respectées)
Catégorie 3	$240 \leq X \leq 510$	+1 pour cette catégorie (limites respectées)

Tableau 3.4 Discrimination de catégorie selon les limites d'une caractéristique

Pour cet exemple, selon le tableau 3.4 la catégorie 1 aurait un poids de 0 et les catégories 2 et 3 un poids respectif de +1. Au terme de l'analyse des caractéristiques, la ou les catégories ayant le plus grand poids sont conservées et la ou les autres catégories sont discriminées.

3.2.1 But recherché par la discrimination de catégories

L'aboutissement de cette étape de discrimination de catégories vise à simplifier et accélérer l'étape suivante qui est la classification d'images. En d'autres mots, moins il y a de catégories possibles pour une image donnée, moins il y a de traitements comparatifs à faire entre les catégories restantes (selon les principes d'un contre un ou un contre tous de Hsu et Lin [2002]). Le choix de la catégorie d'appartenance de l'image traitée (sa classification) se trouve alors simplifié.

3.2.2 Calcul des limites d'une caractéristique

Dans le cadre de cette recherche, le calcul de la limite minimale et maximale d'une caractéristique propre à une catégorie donnée est réalisé lors de l'entraînement du classificateur. L'étape d'entraînement consiste à fournir un groupe d'images ayant été préalablement identifié manuellement comme appartenant à une même catégorie. Chacune des images du groupe passe d'abord par l'étape de prétraitement de l'image (section 3.1) permettant de faire la caractérisation de celles-ci. Ensuite, toutes les caractéristiques sont regroupées par type et assignées à la catégorie d'appartenance. Au terme de ce traitement, chaque catégorie possède une liste de valeurs pour chacune des caractéristiques possibles.

La suite consiste à traiter chaque caractéristique indépendamment afin de calculer la moyenne (\bar{x}) et l'écart-type (σ) de la liste de N valeurs obtenue à l'étape précédente. Pour cette recherche, il fut supposé que chacune des caractéristiques des images suivait une distribution selon la loi normale [Baillargeon, 1990]. C'est pourquoi les limites inférieure X_{min} et supérieure X_{max} furent calculées comme suit :

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad (3.5)$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N - 1}} \quad (3.6)$$

$$X_{min} = \bar{x} - (3\sigma) \quad (3.7)$$

$$X_{max} = \bar{x} + (3\sigma) \quad (3.8)$$

Tel que spécifié dans l'ouvrage de Baillargeon [1990], l'usage de ces formules permet d'avoir une couverture de la distribution de la population de l'ordre de 99.75 %.

3.2.3 Traitement de discrimination

Deux approches possibles permettant la discrimination de catégories ont été considérées pour cette recherche. Soit une première possibilité qui est de comparer chaque caractéristique de l'image traitée avec chacune des limites de l'ensemble des catégories possibles. Et une seconde possibilité, qui est de définir un arbre décisionnel à intervalles pour chacune des caractéristiques possibles selon les intervalles minimaux et maximaux de chaque catégorie (chacune des caractéristiques de l'image traitée doit passer dans chacun des arbres décisionnels appropriés). Au terme des deux approches, la liste des catégories non discriminées (donc ayant le plus grand poids) est transmise au classificateur SVM.

Pour déterminer quelle est la meilleure approche de discrimination de catégories, une analyse du temps de traitement requis par approche a été réalisée. D'abord, le premier choix possible nécessite la comparaison de N caractéristiques d'une image en fonction de M catégories, ce qui donne un nombre de comparaisons possibles par image de $M \times N$ (donc le maximum de traitement possible en tout temps). Tandis que le second choix offre la possibilité de diminuer le nombre de comparaisons nécessaires par caractéristique. Par exemple, un objet d'intérêt ayant une très grande surface (aire) ne devrait pas être comparé avec toutes les catégories ayant une petite surface. Cette comparaison devrait se faire uniquement avec les catégories ayant une large surface. C'est pourquoi l'arbre décisionnel à intervalles a été retenu dans cette recherche pour cette partie du traitement. Tel que décrit dans l'article de Safavian et Landgrebe [1991], l'arbre décisionnel vise à diminuer le nombre de décisions nécessaire à l'atteinte d'une solution désirée, en regroupant les noeuds et/ou feuilles ayant des similitudes dans une même section d'un arbre.

3.2.4 Arbre décisionnel à intervalles

Un arbre décisionnel à intervalles (« Interval Tree ») est une structure organisée sous la forme d'un arbre inversé. Cette structure est composée de noeuds et de feuilles. Chaque noeud a deux enfants et un parent (sauf le noeud racine qui n'a pas de parent) et chaque feuille à un parent, mais n'a pas d'enfant(s). Le parent d'un noeud est le noeud précédent son atteinte et ses enfants sont les noeuds et/ou feuilles au bout de la ou des branches de ce noeud [Safavian et Landgrebe, 1991]. Les figures 3.9 et 3.10 présentent les étapes de création d'un arbre décisionnel à intervalles.

CARACTÉRISTIQUE 1

A)

CATÉGORIE	INTERVALLE [min, max]
Catégorie 1	[26, 26]
Catégorie 2	[16, 21]
Catégorie 3	[5, 8]
Catégorie 4	[8, 9]
Catégorie 5	[25, 30]
Catégorie 6	[15, 23]

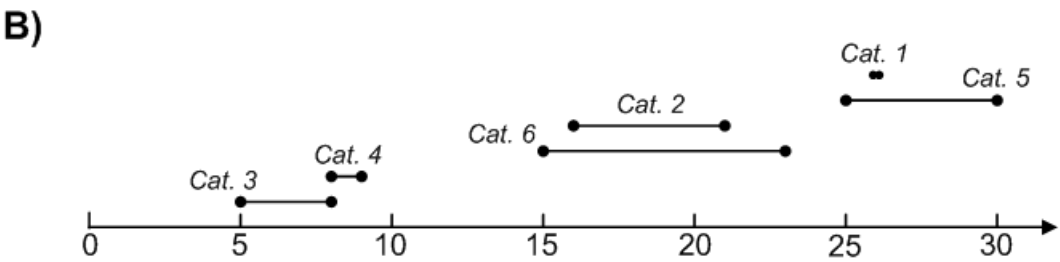


Figure 3.9 Étapes A et B de création d'un arbre décisionnel à intervalles⁶

CARACTÉRISTIQUE 1

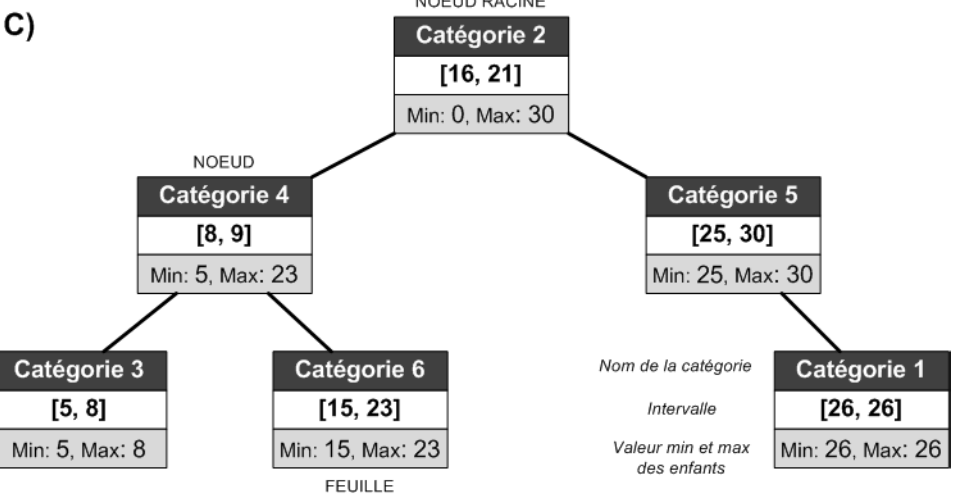


Figure 3.10 Étape C de création d'un arbre à intervalles⁷

La première étape pour créer un arbre à intervalles consiste à regrouper les catégories et leur intervalle en fonction d'une caractéristique donnée A). Ensuite, il faut trier les catégories selon la limite minimale de leur intervalle, de la plus petite valeur à la plus grande B). Finalement, il y a la création de l'arbre à intervalles dont les noeuds et feuilles sont positionnés de gauche à droite en fonction de la limite inférieure de l'intervalle de

⁶Inspiré de [Cormen *et al.*, 1989]

⁷Inspiré de [Cormen *et al.*, 1989]

ces noeuds et feuilles C). Chaque intersection de l'arbre (noeud ou feuille), contient un identifiant qui est le nom de la catégorie représentée, un intervalle qui est la valeur minimale et maximale couverte par la caractéristique de la catégorie, puis une valeur minimale et maximale qui donne l'intervalle de couverture du noeud jumelée avec celles de ses enfants (s'il y a lieu) [Cormen *et al.*, 1989].

Un exemple d'utilisation de l'arbre à intervalles de la figure 3.10 est décrite ci-après. Supposons qu'une image X a 26 comme valeur pour sa caractéristique 1. Donc la séquence de traitement sera la suivante :

1. Selon la valeur minimale et maximale du noeud racine est ce que la valeur 26 est à l'intérieure de l'intervalle ? Oui, $[0, 30]$ donc on poursuit l'analyse.
2. Selon le noeud actuel (racine), est-ce que 26 est à l'intérieur de son intervalle ? Non, cette intervalle étant $[16, 21]$ cette catégorie est rejetée.
3. Est-ce 26 est à l'intérieur de la valeur minimale et maximale de l'enfant de **gauche** de ce noeud ? Non, $[5, 23]$ donc aucun passage à gauche.
4. Est-ce 26 est à l'intérieur de la valeur minimale et maximale de l'enfant de **droite** de ce noeud ? Oui, $[17, 30]$ donc passage à l'enfant de droite.
5. Selon le noeud actuel, est-ce que 26 est à l'intérieur de son intervalle ? Oui, étant $[25, 30]$ le poids de la catégorie 5 est augmenté de 1 (+1).
6. La suite consiste à répéter la séquence à partir de l'étape 3, jusqu'à ce qu'il n'y ait plus d'intervalles respectant la valeur analysée.

Tel que cité précédemment, un arbre décisionnel à intervalles doit être créé pour chacune des caractéristiques utilisées et le total final du poids de chaque catégorie, résultant du passage dans tous les arbres, détermine les catégories étant transférées au classificateur (voir la section 3.3).

3.3 Classification

Cette étape finale du processus de classification d'une image consiste à déterminer la catégorie à laquelle l'image de phytoplancton appartient. Telle que définie précédemment dans ce mémoire de recherche, la méthode de classification SVM a été choisie pour le projet *Biotaxis*.

Le développement d'un classificateur à généralement pour but de répondre à un problème spécifique. L'objectif premier est de classer correctement différents objets, et ce, avec la plus grande précision possible [Abe, 2010]. L'idée directrice est donc de définir un ou des modèles décisionnels permettant d'obtenir la meilleure approximation de classification pour l'ensemble des objets observés.

3.3.1 Classification SVM linéaire

Le cas le plus simple pour l'élaboration d'un classificateur est lorsque l'on peut appliquer un séparateur linéaire. Donc, lorsqu'il est possible de séparer les 2 classes d'un univers de données \mathbb{R}^n par un hyperplan linéaire. Le livre de Hamel [2009] décrit bien cette problématique et l'encadré, présenté ci-après, est une traduction d'un modèle général de représentation de la classification SVM linéaire décrit dans ce volume⁸ :

- Soit l'univers de données \mathbb{R}^n , avec les vecteurs $\vec{x} \in \mathbb{R}^n$ représentant les objets.
- Un échantillon de données S , où $S \subset \mathbb{R}^n$.
- Une fonction cible $f : \mathbb{R}^n \rightarrow \{+1, -1\}$.
- Un groupe de formation pré identifié $D = \{(\vec{x}, y) \mid \vec{x} \in S \text{ et } y = f(\vec{x})\}$.

Sur cette base, il faut déterminer une fonction $\hat{f} : \mathbb{R}^n \rightarrow \{+1, -1\}$ en utilisant D de sorte que : $\hat{f}(\vec{x}) \cong f(\vec{x})$, pour tous $\vec{x} \in \mathbb{R}^n$ (donc que le modèle \hat{f} soit une bonne approximation de la fonction f).

Cet ensemble d'informations requiert de plus amples explications :

- D'abord, l'univers de données est représenté par l'espace \mathbb{R}^n , où n représente les différents plans de cet univers.
- L'échantillon de données S est constitué d'un groupe de données devant représenter adéquatement l'ensemble des données de l'univers. Leur sélection doit être faite de façon à ce que chacune des catégories possibles ($\{+1, -1\}$ dans le cas présent) soit bien définie.
- Le modèle \hat{f} définit la surface décisionnelle qui est représentée par une ligne, un plan ou un hyperplan séparant les différentes classes dans l'univers \mathbb{R}^n .
- Dans le cas plus complexe d'une classification de classes multiples (multiclasses), les valeurs en sortie $\{+1, -1\}$ sont alors remplacées par l'ensemble des valeurs possibles soient $\{1, 2, \dots, M\}$.

⁸Le texte dans l'encadré est une traduction de l'anglais dont le contenu provient du livre de Hamel [2009].

Note : lorsqu'un groupe de formation préidentifiée D est utilisé pour définir un modèle \hat{f} , alors le modèle sera dit basé sur un apprentissage supervisé. Inversement, s'il n'y a pas de groupe de formation D utilisé alors l'apprentissage du modèle sera non supervisé.

En partant du groupe de formation prédéfini D , il est possible de projeter sur un graphique l'ensemble des vecteurs \vec{x} qui le composent afin de trouver le modèle \hat{f} . Voici l'exemple d'un graphique bidimensionnel représentant cette projection selon un classificateur SVM linéaire :

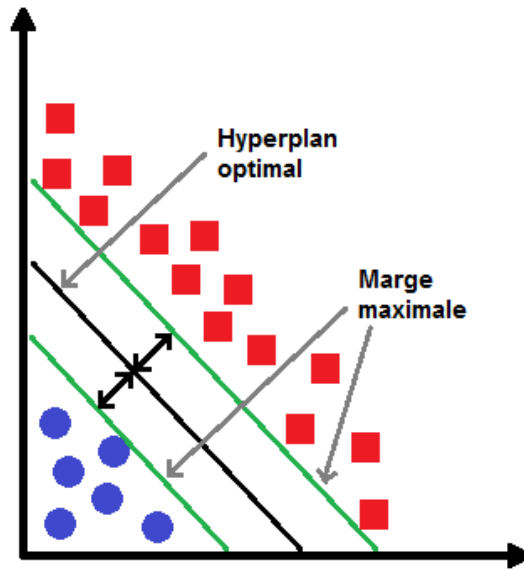


Figure 3.11 Classificateur SVM avec hyperplan optimal et marge maximale

Le graphique de la figure 3.11 présente des informations nécessitant une attention particulière. Pour débiter, on retrouve un ensemble de points représenté par deux formes (cercle et carré). Les cercles peuvent être remplacés par $y_i = +1$ et les carrés par $y_i = -1$, selon l'équation $D = \{(\vec{x}, y) \mid \vec{x} \in S \text{ et } y = f(\vec{x})\}$ définie précédemment. Ensuite on retrouve deux hyperplans de support (marge maximale), qui sont des traits parallèles distants de $\frac{2}{\|\vec{w}\|}$. Ceux-ci définissent les limites maximales de séparation entre les cercles (+1) et les carrés (-1). à partir des équations représentant ces hyperplans, il est possible d'obtenir la forme générale suivante :

- Sachant que $\vec{w} \cdot \vec{x}_i - b \geq +1$
- et que $\vec{w} \cdot \vec{x}_i - b \leq -1$
- alors on obtient la formule générale : $y_i(\vec{w} \cdot \vec{x}_i - b) - 1 \geq 0$, pour $i = 1, \dots, M$.

⁹Le texte dans l'encadré est une traduction de l'anglais dont le contenu provient du livre de Hamel [2009].

Le tracé nommé hyperplan optimal qui divise uniformément les deux hyperplans de support est représenté par l'équation $\vec{w} \cdot \vec{x} - b = 0$, qui représente la surface maximale de décision. Le but recherché par l'ensemble de ce processus est de définir un hyperplan permettant d'obtenir la plus grande précision de classification possible.

3.3.2 Classification SVM non linéaire

Après la classification SVM linéaire vient la classification SVM non linéaire. Cette dernière survient lorsque le classificateur ne peut pas être défini directement par un hyperplan linéaire. Pour être en mesure de concevoir un classificateur non linéaire, il faut utiliser le « *kernel trick* » [Hamel, 2009]. L'idée de cette approche est de transformer un ensemble de données non séparable linéairement en un espace dimensionnel supérieur, appelé espace fonctionnel (*feature space*), où les données peuvent être séparées linéairement [Hamel, 2009]. La figure 3.12 illustre le but recherché par l'application du « *kernel trick* » :

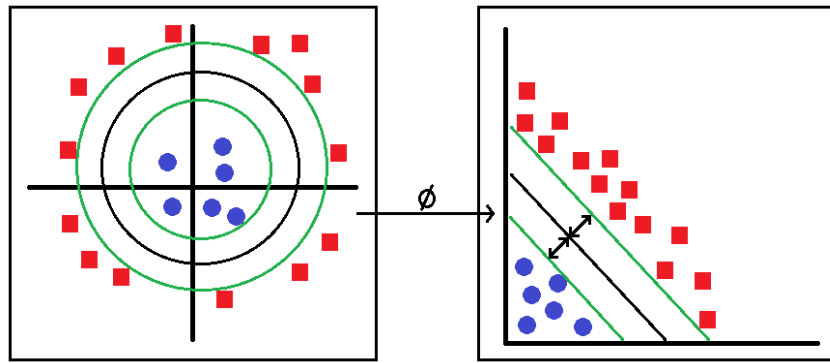


Figure 3.12 Mappage d'un problème non linéaire vers un espace supérieur

En partant de la fonction décisionnelle \hat{f} décrite précédemment et en la reconstruisant selon le mappage de l'espace initial vers l'espace fonctionnel Φ , on obtient alors [Hamel, 2009] :

$$\hat{f}(\vec{x}) = \text{sgn}(\vec{w}^* \cdot \Phi(\vec{x}) - b^*) \text{ avec } \Phi : \mathbb{R}^n \rightarrow \mathbb{R}^m, \text{ où } m \geq n \quad (3.9)$$

Le vecteur normal \vec{w}^* de cette fonction décisionnelle peut-être représenté par la dualité suivante : $\vec{w}^* = \sum_{i=1}^l \alpha_i^* y_i \Phi(\vec{x}_i)$. En remplaçant \vec{w}^* par sa dualité dans l'équation 3.1, on obtient la fonction décisionnelle :

$$\hat{f}(\vec{x}) = \text{sgn}\left(\sum_{i=1}^l \alpha_i^* y_i \Phi(\vec{x}_i) \cdot \Phi(\vec{x}) - b^*\right) \quad (3.10)$$

Par définition la représentation d'une fonction *kernel* est donnée par $k(\vec{x}, \vec{y})$, où $\vec{x}, \vec{y} \in \mathbb{R}^n$. Puisque $k(\vec{x}, \vec{y}) = \Phi(\vec{x}) \cdot \Phi(\vec{y})$, on peut réécrire la fonction décisionnelle en tenant compte

de la fonction *kernel*, ce qui donne :

$$\hat{f}(\vec{x}) = \text{sgn}(\sum_{i=1}^l \alpha_i^* y_i k(\vec{x}_i, \vec{x}) - b^*) \quad (3.11)$$

Cette série d'étapes mathématiques représente l'application du « *kernel trick* ». Soit d'appliquer un *kernel* à la fonction décisionnelle afin de faire le mappage direct des données vers l'espace fonctionnel. La décision restante est le choix du *kernel* à employer, soit d'établir celui qui via le mappage sépare le mieux les différentes classes afin de pouvoir les subdiviser par un hyperplan linéaire. Le tableau 3.5 présenté ci-après, énumère quelques-uns des *kernels* pouvant être utilisés :

Nom du <i>kernel</i>	Fonction du <i>kernel</i>
<i>Kernel</i> linéaire	$k(\vec{x}, \vec{y}) = \vec{x} \cdot \vec{y}$
<i>Kernel</i> polynomial homogène	$k(\vec{x}, \vec{y}) = (\vec{x} \cdot \vec{y})^d$
<i>Kernel</i> polynomial non homogène	$k(\vec{x}, \vec{y}) = (\vec{x} \cdot \vec{y} + c)^d$
<i>Kernel</i> gaussien	$k(\vec{x}, \vec{y}) = e^{-(\ \vec{x} - \vec{y}\ ^2 / 2\beta^2)}$

Tableau 3.5 Quelques *kernels* fréquemment utilisés¹⁰

Il est important de souligner que parmi les différents articles référés traitants de la classification d'images de phytoplancton, nombreux sont ceux qui ont utilisé le *kernel* gaussien (aussi appelé *radial basis function kernel* ou *RBF kernel*) avec la méthode SVM [Alvarez *et al.*, 2012; Luo et Kramer, 2004; Sosik et Olson, 2007; Su *et al.*, 2010; Verikas *et al.*, 2012]. Dans le cadre de cette recherche, c'est le *kernel* gaussien qui a été employé. Ce choix étant appuyé par la citation de son usage dans les articles cités ci-haut et dû à l'article de Hsu *et al.* [2010] qui présente des motifs justifiant son utilisation (voir l'encadré ci-après).

L'article de Hsu *et al.* [2010], qui est un guide sur la classification SVM, cite que le *RBF kernel* est un premier choix raisonnable dû aux raisons suivantes :

- Il peut être appliqué aux classes ayant une relation non linéaire (impossible avec le *kernels* linéaire).
- Il a moins de paramètres à configurer que les *kernels* polynomiaux (donc moins complexe que ceux-ci).
- Il donne des valeurs comprises entre 0 et 1, comparativement aux *kernels* polynomiaux dont les valeurs varient entre 0 et l'infini positif (présentant moins de difficultés numériques).

¹⁰source [Hamel, 2009]

3.3.3 Traitement de classes multiples

Il arrive parfois que l'on ait à effectuer la classification de classes multiples. L'une des approches les plus souvent utilisées consiste à décomposer le problème en plusieurs sous-problèmes de classification binaire. Cette représentation binaire peut se faire de deux façons différentes [Hamel, 2009] :

- La première option se nomme *un contre tous* (one-vs-all). Cette méthode consiste à séparer chaque classe du reste des autres classes. Ainsi pour un nombre de classes possible M , il y aura alors M classifications binaires à effectuer.
- La deuxième option se nomme *un contre un* (one-vs-one). Cette dernière consiste à établir toutes les combinaisons possibles de jumelage par pair entre chacune des classes disponibles. L'usage de cette approche requiert un total de $M(M - 1)/2$ classifications binaires.

L'étude de Hsu et Lin [2002] sur les deux méthodes énoncées ci-haut a démontré que dans de nombreuses situations, l'approche *un contre un* était plus rapide et plus précise que l'approche *un contre tous* lors de la résolution d'un problème de classification SVM. Luo et Kramer [2004] ont fait des tests similaires avec des images de phytoplancton et sont arrivés à la même conclusion que Hsu et Lin, c'est-à-dire que l'approche *un contre un* est préférable à l'approche *un contre tous*.

3.4 Précision et temps de traitement

À titre de rappel, les objectifs visés par cette recherche sont d'obtenir une précision de la classification de 87 % et un temps moyen de classification de 200 ms par image, ce qui équivaut à 5 images par seconde. Il est donc important d'élaborer sur les méthodes employées pour le calcul de ces deux valeurs.

Pour obtenir la valeur de la précision de la classification l'article de Fawcett [2006] a été utilisé comme référence. La figure 3.13 présente de quelle façon la précision est calculée.

		<u>True class</u>		
		p	n	
<u>Hypothesized class</u>	Y	True Positives	False Positives	$\text{precision} = \frac{TP}{TP+FP}$ $\text{recall} = \frac{TP}{TP+FN}$
	N	False Negatives	True Negatives	
Column totals:		P	N	

Figure 3.13 Matrice de confusion et calcul de la précision¹¹

Dans la figure 3.13, trois types d'assignation à une image classifiée requiert des explications pour pouvoir passer par la suite au deux équations présentes dans cette même figure. Ces trois types d'assignation sont les suivants :

- Le terme TP qui représente l'« True Positive » (Vrai Positif) désigne une image dont on connaît la catégorie d'appartenance et qui a été classifiée dans la bonne catégorie.
- FN qui signifie « False Negative » (Faux négatif) représente l'assignation d'une catégorie autre que la catégorie réelle à une image donnée.
- Si une image est classifiée dans une autre catégorie que sa catégorie d'appartenance, alors celle-ci sera comptée comme « False Positive » FP (Faux Positif).
- Finalement, le terme TN représente une image n'ayant pas été assignée à une mauvaise catégorie.

Dans le contexte présent, différents groupes d'images *test*, dont la catégorie d'appartenance était déjà connue, a été utilisée pour calculer la précision du classificateur. En se basant sur la formule du calcul de la précision énoncé dans la figure 3.13, la précision a été calculée par catégorie selon la moyenne de la précision de la classification de l'ensemble des images d'une catégorie. Puis globalement, par le calcul de la moyenne de la précision de la classification selon la valeur de la précision de l'ensemble des catégories.

La formule de précision de la figure 3.13 représente le ratio de bonne classification (TP) versus la sommation de bonne (TP) et de mauvaise classification dans des catégories autres que la catégorie réelle (FP).

¹¹Source [Fawcett, 2006]

La seconde formule désignée avec le terme « Recall » (qui est aussi appelée « Hit Rate » ou ratio de classification), représente le ratio de bonne classification (TP) versus la sommation de bonne (TP) et mauvaise classification pour une catégorie donnée (FN).

Pour ce qui est du calcul du temps de traitement requis par classification, un calculateur de temps a été ajouté au logiciel *Biotaxis*. Le principe de fonctionnement du calculateur utilisé se présente comme suit (en Java) :

```
long startTimer = System.currentTimeMillis();  
//Exécution d'opération(s) ...  
long totalTime = System.currentTimeMillis() - startTimer;
```

Ainsi, cette séquence permet d'obtenir une approximation du temps de traitement requis (en millisecondes) pour exécuter une suite d'opérations (tel que la classification d'une image). Le terme *approximation* signifie que la valeur obtenue est une représentation du temps nécessaire à l'exécution de la suite d'opération, car cette valeur est influencée par le temps de partage des ressources de l'ordinateur utilisé avec les autres applications en cours de fonctionnement. Par exemple, s'il y a un processus requérant beaucoup de ressource mémoire est en cours de fonctionnement sur l'ordinateur et qu'une classification est en cours, le temps requis de traitement pourrait être plus long que si ce même processus était en arrêt.

Cette partie sur la précision et le temps de traitement complète la section de l'état de l'art. La section suivante sur le développement de cette recherche présente de quelle façon les concepts de l'état de l'art ont été appliqués au projet *Biotaxis*.

3.5 Résumé de l'état de l'art

Ce chapitre sur l'état de l'art a d'abord servi à l'énonciation des avancées ayant été recensées dans la littérature, dans le but de définir les méthodes et/ou approches employées dans cette recherche. Celui-ci a débuté par un rappel des objectifs ciblés afin de formuler l'orientation que cette étude littéraire a empruntée. Ces objectifs étant de réaliser un logiciel de classification, dont l'évaluation sera basée sur la qualité de la précision de la classification et sur le temps de traitement requis. Ainsi, la sélection des méthodes de traitement, ayant été présentée dans l'état de l'art, fut orientée en vue d'atteindre les objectifs ciblés.

L'ordre de présentation des études sur les étapes de traitement employées fut établi en fonction des deux solutions de classification proposées et citées au début de l'état de

l'art (voir les schémas de la figure 2.1). On retrouve d'abord une analyse de la phase de prétraitement qui comprend dans l'ordre les étapes : de filtrage et de détection de contour avec les méthodes de Sobel et Feldman [1968] et de Prewitt [1970] ; de seuillage selon la méthode hystérésis [Pridmore, 2002] ; de tracé de contour de l'élément principal avec la méthode *Convex Hull* [de Berg *et al.*, 2008] ; et de caractérisations de l'élément principal selon le *Convex Hull* [Luo, 1998]. Ensuite, on retrouve la phase de discrimination de catégories, qui présente l'arbre décisionnel à intervalles (basé sur l'ouvrage de Cormen *et al.* [1989]), puis la phase de classification d'images selon la méthode SVM [Hamel, 2009]. Une fois la classification d'image complétée, une analyse de la qualité de cette classification est effectuée. Les critères d'évaluation de la qualité sont présentés à la section sur la précision de la classification [Fawcett, 2006] et du temps de traitement requis.

Les phases de traitement et les critères d'évaluation, qui ont été étudiés et énoncés dans ce chapitre, sont à la base des algorithmes de traitement qui ont été implémentés et intégrés au logiciel de classification *Biotaxis*. Les détails, relatifs à la réalisation et à l'intégration des phases de traitement au logiciel, sont donnés dans le chapitre suivant qui traite du développement logiciel (voir chapitre 4).

CHAPITRE 4

DÉVELOPPEMENT LOGICIEL

Pour ce projet de recherche, l'un des trois objectifs secondaires visé a été de développer un logiciel de classification d'images, afin de permettre la réalisation des deux autres objectifs ciblés, étant :

- d'atteindre une précision moyenne de la classification d'image d'environ 87 % (à +/- 1 % d'écart) ;
- d'obtenir une moyenne minimale de temps de traitement par classification d'image de 200 ms (soit de pouvoir traiter 5 images par seconde).

L'outil logiciel *Biotaxis*, représente la plateforme de test et de validation des concepts et principes qui ont été à l'étude, tout en étant un outil permettant de vérifier l'atteinte des objectifs cité ci-haut. Cette section sur le développement du projet de recherche met l'emphasis sur les étapes qui ont été appliquées pour concevoir le logiciel *Biotaxis*.

Ce chapitre sur le développement de la recherche est subdivisé en quatre sections, soit :

- L'analyse préliminaire.
- La conception du logiciel *Biotaxis*.
- Les tests du logiciel.
- Comment classifier une image avec le logiciel *Biotaxis*.

Ce chapitre fait donc le lien entre l'énoncé des théories et concepts cités dans le chapitre de l'état de l'art, avec leur mise en application dans un contexte pratique grâce aux fonctionnalités du logiciel *Biotaxis*. Les résultats obtenus à l'aide de ce logiciel ainsi que leur analyse sont présentés dans le chapitre suivant.

4.1 Analyse préliminaire

Cette première partie du développement, qui est l'analyse préliminaire, traite des différents points importants ayant requis une attention particulière en début de projet. Les solutions présentées dans cette section ont permis de mieux situer les étapes et requis nécessaire au démarrage, ainsi qu'au bon déroulement du développement de projet.

4.1.1 Quelles images utiliser pour la classification ?

La première interrogation soulevée par cette recherche fut de déterminer quelles images seraient à classer. Pour mener à bien un projet de recherche sur la classification d'images, un ou des groupes d'images préclassifiées sont nécessaires. Tel que spécifié à la fin de l'introduction, une série d'images provenant d'un appareil IFCB et ayant été utilisées pour la recherche de Sosik et Olson [2007], furent disponibles pour mener à bien cette recherche. Ces images ont l'avantage d'avoir été préclassifiées et subdivisées en deux groupes distincts, soit un premier groupe servant à l'entraînement du classificateur et un second, pour tester la précision de la classification de ce dernier. Chacun de ces groupes est composé de 3 150 images de phytoplancton et est subdivisé en 21 catégories distinctes de 150 spécimens. La figure 4.1 présente un exemple d'image de chacune des 21 catégories de phytoplancton ayant servi à la réalisation de cette recherche.

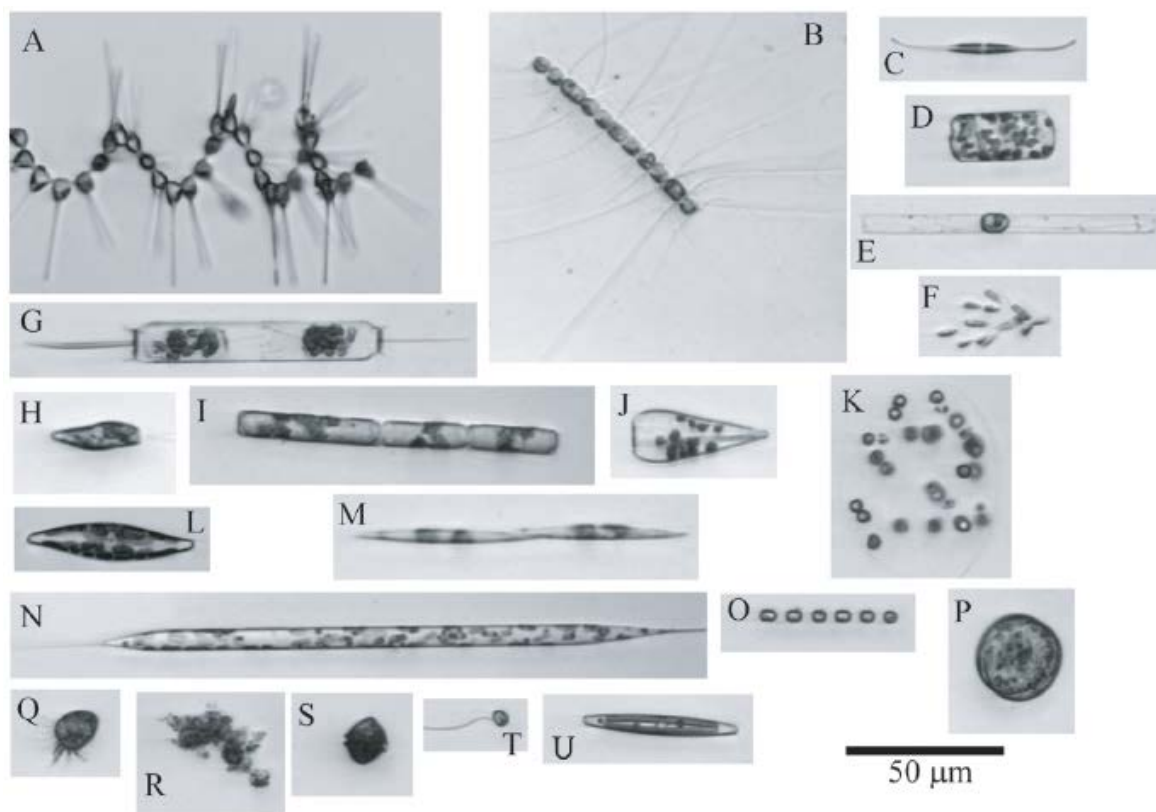


Figure 4.1 Exemple d'images des 21 catégories de phytoplancton utilisées pour cette recherche¹

Les noms des espèces de phytoplanctons présentées dans la figure 4.1 sont : (A) *Asterionellopsis*; (B) *Chaetoceros*; (C) *Cylindrotheca*; (D) *Cerataulina*; (E) *Dactyliosolen*;

¹Source [Sosik et Olson, 2007]

(F) *Dinobryon*; (G) *Ditylum*; (H) *Euglena*; (I) *Guinardia*; (J) *Licmophora*; (K) *Phaeocystis*; (L) *Pleurosigma*; (M) *Pseudonitzschia*; (N) *Rhizosolenia*; (O) *Skeletonema*; (P) *Thalassiosira*; (Q) *Ciliate*; (R) *Detritus*; (S) *Dinoflagellates*; (T) *Nanoflagellates*; (U) *Pennate*.

Le fait d’avoir accès à deux groupes d’images préclassifiées de 21 catégories distinctes de phytoplancton offre plusieurs avantages, tels que :

- Éviter d’avoir à classifier manuellement un grand nombre d’images de phytoplancton.
- Offrir une diversité de catégories (soit 21), offrant une certaine marge de manoeuvre sur le choix de la séquence d’étapes menant à l’atteinte des objectifs (voir le chapitre 5 « Résultats et Analyse » pour plus de détails).
- Obtenir un temps de traitement moyen de la classification selon des groupes d’images de dimensions diverses (la dimension de l’image ayant un impact direct sur le temps de prétraitement, tel que le balayage avec un filtre ou l’acquisition des caractéristiques via le *Convex Hull*).
- Pouvoir entraîner et tester le classificateur du logiciel *Biotaxis* avec des groupes d’images distincts, donc éviter d’avoir des résultats biaisés.

Bien entendu, cette liste d’avantages n’est pas exhaustive, mais elle présente les avantages majeurs ayant mené à l’adoption de ces groupes d’images pour réaliser ce projet de recherche.

En ce qui a trait à l’organisation des fichiers images, celles-ci ont été séparées et insérées dans deux dossiers distincts. Il y a un premier dossier comprenant un ensemble d’images d’entraînement et un second ayant des images de test. La figure 4.2, présentée ci-après, illustre la structure organisationnelle d’un ensemble d’images.

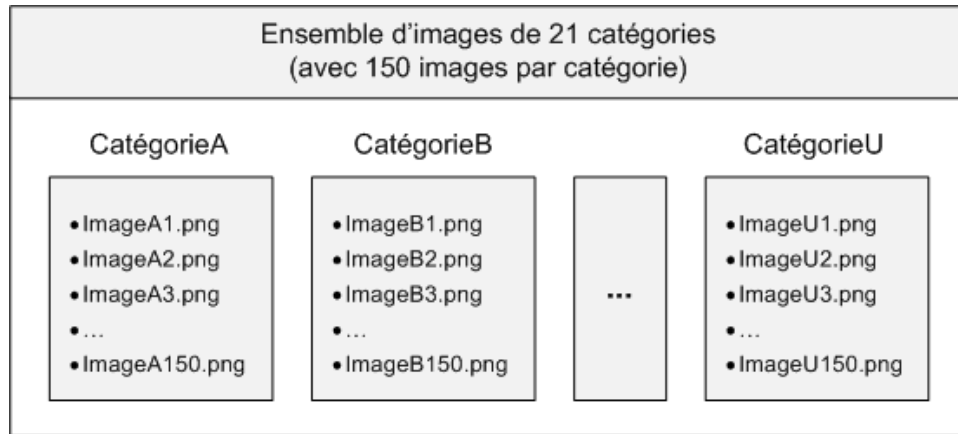


Figure 4.2 Organisation des fichiers images en ensemble d'images de 21 catégories

4.1.2 Identification des étapes de traitement requises par le logiciel

Dans le chapitre sur les concepts de base, deux séquences d'opérations possibles menant à la classification d'images ont été présentées (voir la figure 2.1). L'analyse de ces informations jumelées avec celles présentées dans le chapitre 3 sur l'état de l'art ont permis d'identifier les deux séquences de traitements requis pour permettre la classification d'images via un outil logiciel.

Entraîner un classificateur

La première séquence consiste à entraîner le classificateur pour créer un groupe de classification². Celle-ci requiert la série d'étapes suivantes (qui est résumé à la figure 4.3) :

1. Créer un groupe de classification.
2. Choisir les catégories qui seront classées par ce groupe.
3. Charger l'ensemble des images d'entraînement des catégories choisies.
4. Démarrer le cycle d'entraînement du groupe de classification.
5. Mise en mémoire d'une image d'entraînement.
6. Acquisition des paramètres de base de l'image (dimension, histogramme, moyenne du niveau de gris, etc.).
7. Prétraitement de l'image.
 - (a) Application du filtre sélectionné à l'image (s'il y a lieu).

²Un groupe de classification représente un ensemble de catégories d'images dont les intervalles de chacune des caractéristiques des catégories ont été calculés.

- (b) Seuillage de l'image avec la méthode hystérésis.
 - (c) Tracé de contour de l'élément principal avec la méthode *Convex Hull*.
 - (d) Acquisition des caractéristiques de l'image.
8. Enregistrer chacune des valeurs caractérisant l'image dans la liste de caractéristiques de sa catégorie d'appartenance.
 9. Répéter à partir de l'étape 5, jusqu'à ce que l'ensemble des images d'entraînement soit traité.
 10. Génération des arbres décisionnels à intervalles (pour faire la discrimination de catégories).
 - (a) Calcul des intervalles (min, max) de l'ensemble des caractéristiques pour toutes les catégories du groupe choisi.
 - (b) Créer une liste pour chaque caractéristique contenant les limites de chaque catégorie pour la caractéristique désignée.
 - (c) Trier chaque liste en fonction de la limite minimale de chaque catégorie (plus petite à la plus grande).
 - (d) Générer un arbre décisionnel à intervalles pour chaque type de caractéristique.
 11. Produire le modèle de classification SVM global³, basé sur l'ensemble des catégories possibles du groupe.
 12. Arrêt du processus d'entraînement du groupe.

³Un modèle de classification SVM est produit à l'aide d'un ensemble d'images d'entraînement, afin de prédire la catégorie d'appartenance d'une image test en fonction des caractéristiques de cette image (traduction de l'article de Hsu *et al.* [2010]).

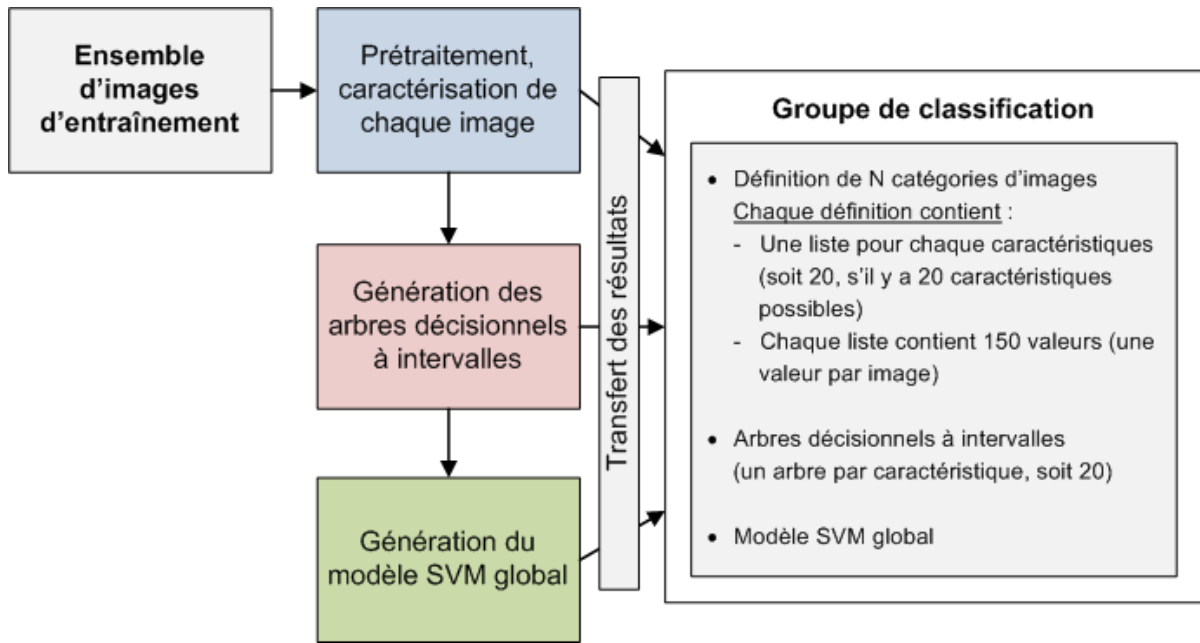


Figure 4.3 Séquence d'étapes d'entraînement d'un ensemble d'images menant à la création d'un groupe de classification

Tester un classificateur

La deuxième séquence vise à tester la précision et le temps de traitement moyen d'un groupe de classification ayant été préalablement entraîné. Cette série d'étapes se présente comme suit (voir la figure 4.4 pour un aperçu général) :

1. Choisir le groupe de classification désiré. *Cette sélection amène le chargement des paramètres liés au groupe qui sont ; la liste de catégories d'images du groupe, les caractéristiques de chacune des catégories, les arbres décisionnels à intervalles et le modèle SVM global.*
2. Sélectionner une ou des images préclassifiées qui serviront à tester le groupe de classification. *Il est important que la catégorie des images tests sélectionnées soit présente dans la liste de catégorie du groupe, autrement le classificateur assignera la catégorie ayant le plus de caractéristiques similaires avec l'image testée, sans possibilité d'assigner la bonne catégorie étant donné que celle-ci n'est pas couverte par le groupe (le chapitre 6 présente une solution qui pourrait-être envisagée pour ce cas particulier dans un projet futur).*
3. Activer ou non la discrimination d'image (optionnel selon le choix de la solution désirée, soit 1 ou 2).
4. Démarrer le cycle de test d'image(s) avec le groupe de classification.

5. Mise en mémoire d'une image test.
6. *Exécuter la même séquence d'acquisition et de prétraitement d'image que pour l'entraînement (étapes 6 et 7a à 7d).*
7. Solution 1 : Avec discrimination.
 - (a) Faire le passage de chaque caractéristique de l'image testée, dans l'arbre décisionnel à intervalles adéquat.
 - (b) Compilation de la liste de catégories non discriminées par les arbres.
 - (c) Produire un modèle de classification SVM en fonction de la liste de catégories non discriminées.
 - (d) Sélectionner ce modèle pour effectuer la classification.
8. Solution 2 : Sans discrimination.
 - (a) Sélectionner le modèle de classification SVM global pour effectuer la classification (ce modèle étant obtenu lors de l'entraînement du groupe de classification).
9. Classification de l'image avec le modèle SVM choisi (abouti à l'attribution d'une catégorie à l'image testée).
10. Calcul du temps de traitement de l'image testée (temps requis pour effectuer la classification).
11. Évaluation de la précision de la classification pour cette image (comparaison entre la catégorie réelle versus celle attribuée).
12. Compilation statistique de la précision et du temps de traitement selon la catégorie réelle et celle attribuée.
13. Répéter à partir de l'étape 5, jusqu'à ce que l'ensemble des images soit testé.
14. Arrêt du processus de test du groupe de classification.

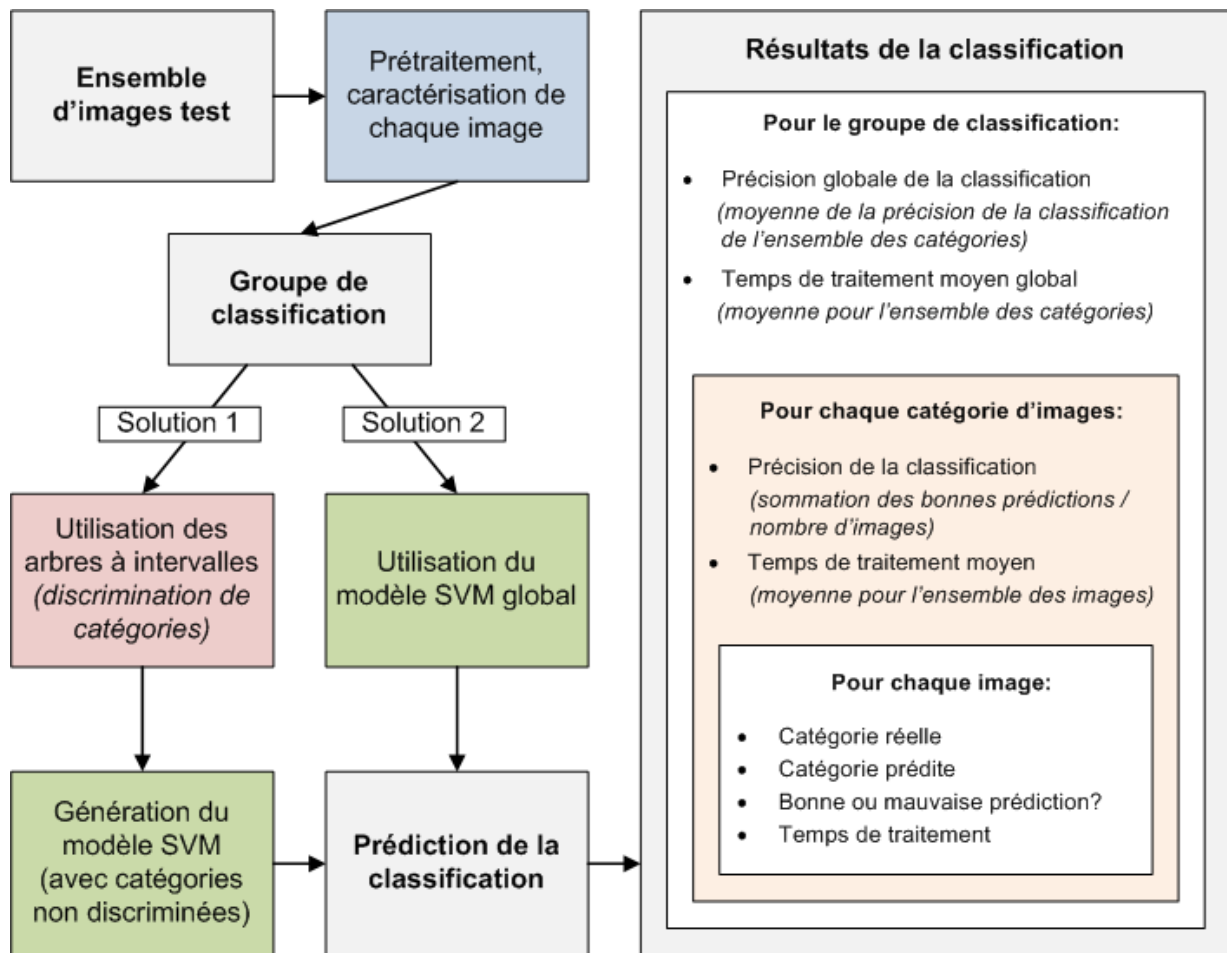


Figure 4.4 Séquence d'étapes de test d'un ensemble d'images et d'un groupe de classification menant à la classification d'images (comprend l'intégration de la séquence d'étapes selon les solutions 1 et 2).

Dans le cas de l'entraînement de groupes de classification, plus il y aura d'images utilisées, plus grande sera la représentativité des catégories présentes dans le groupe de classification. Pour ce qui est du test des groupes de classification, un grand nombre d'images permettra d'avoir une moyenne de la précision et du temps de traitement plus représentatifs de la population globale de chaque catégorie.

4.1.3 Choix des langages de programmation et du logiciel de développement

Plusieurs langages de programmation peuvent-être utilisé pour développer le logiciel. Avant de faire un choix final, plusieurs questionnements se devaient d'avoir une réponse. Par exemple :

- Quelles sont les connaissances déjà acquises sur le langage choisi ?

- Est-ce que la documentation est facilement accessible et couvre plusieurs sujets ?
- Y a-t-il un coût lié à l'usage du langage choisi (performance, temps de traitement) ?
- Y a-t-il des restrictions à son usage ?

Suite à ces questionnements, il fut décidé de développer le logiciel *Biotaxis* à l'aide de deux langages de programmation, soit le Java et le C++. Tout d'abord, le choix du langage Java fut avant tout de l'utiliser pour la conception de l'interface utilisateur du logiciel. Les autres motifs étant qu'il est [Java, 2012] :

- Rapide d'implémenter des applications dans ce langage (ayant un temps de réalisation de projet limité).
- Identique au langage utilisé pour le développement de la librairie libre d'accès *jlibsvm* [Soergel, 2012]. Cette librairie rend possible la classification SVM avec des classes multiples et est dérivée de la librairie *libsvm* [Chang et Lin, 2011] qui fut utilisée pour la recherche de [Sosik et Olson, 2007].
- Facilement portable, car les logiciels développés en Java fonctionnent à l'aide d'une machine virtuelle (JVM), ce qui permet leur exécution sur plusieurs systèmes d'exploitation (tel que Windows, Mac-OS et Linux).
- Libre d'accès, donc aucuns frais d'utilisation, de développement et d'exploitation ne sont encourus.
- Bien documenté et plusieurs exemples d'implémentations sont disponibles.

En ce qui a trait au choix du langage C++, cette décision vient du fait qu'il :

- S'exécute rapidement [Reddy, 2011] (temps de traitement réduit).
- Est possible d'utiliser la librairie JNI (« Java Native Interface ») pour communiquer entre le langage Java et C++ [Liang, 1999].
- Offre la possibilité d'être, éventuellement, intégré dans un système embarqué.

Suite à ces deux choix de langage de programmation, choisir le logiciel de développement *Eclipse* fut aisé [Eclipse, 2013]. Ce dernier est libre d'accès et permet le développement de projet en Java et en C++ sous un même environnement de travail. Le fait d'avoir de l'expérience dans l'utilisation du logiciel *Eclipse* de a également contribué à son adoption pour ce projet de recherche.

4.2 Conception du logiciel *Biotaxis*

Suite à l'analyse préliminaire, il fut possible de réaliser un schéma de l'architecture à haut-niveau (voir la figure 4.5). Ce schéma fait le lien entre les étapes de traitements identifiées à la sous-section 4.1.2 et les langages de programmation utilisés (sous-section 4.1.3).

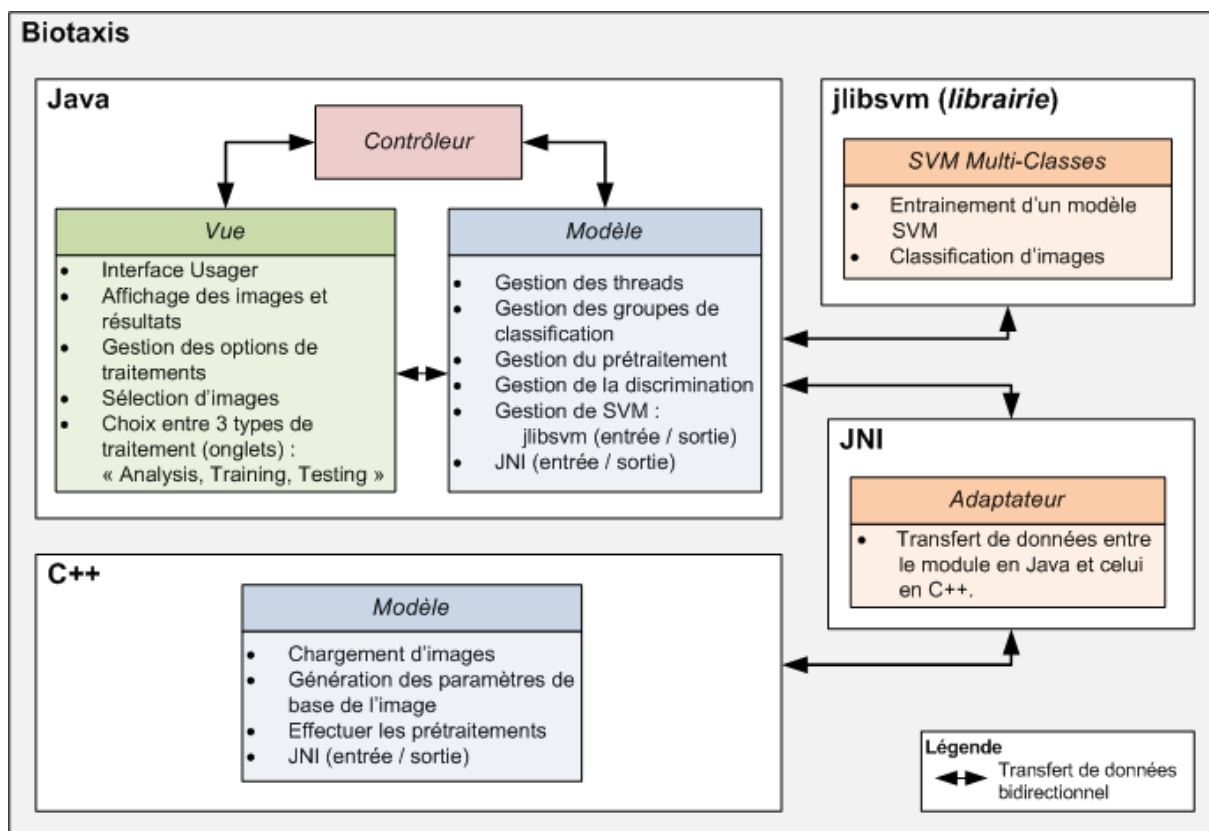


Figure 4.5 Schéma de concept simplifié du logiciel *Biotaxis*

Note : l'architecture logiciel utilisé pour le développement de Biotaxis et présenté à la figure 4.5 est basé sur le modèle de conception MVC (Modèle - Vue - Contrôleur) [Bates et al., 2004]. Cette approche permet de bien découper les différents modules du logiciel selon leurs tâches respectives.

Les sous-sections suivantes, traitant de la conception de *Biotaxis*, font toutes référence au schéma de la figure 4.5. Celui-ci présente un aperçu global du logiciel et vise à faciliter la compréhension des différents modules et segments de traitement qui le compose.

4.2.1 Communication entre les langages Java et C++

La première étape de la conception du logiciel fut de pouvoir établir des échanges de données entre un module implémenté dans le langage Java, avec un second module implé-

menté en C++. Grâce à la librairie JNI (sous-section 4.1.3), un code Java peut appeler les fonctionnalités d'une application développée dans un langage natif (tel que le C++) et inversement, être appelé par cette application [Liang, 1999].

L'ouvrage de Liang [1999] présente plusieurs exemples d'utilisation et d'implémentation de code en Java et C++ communiquant via la librairie JNI. Cependant, une attention particulière doit être appliquée lorsque des objets volumineux, tels qu'une image, sont transférés entre deux applications via JNI. Dans un tel contexte, il est préférable de segmenter l'objet en plusieurs paquets de N kilo-octets et de faire le transfert des données par le biais d'un *Array* de *ByteArray(s)*.

Dans cette recherche, ce sont des images en niveau de gris qui sont utilisées. Ces images ont donc un seul niveau de couleur (trois niveaux équivaut à Rouge, Vert, Bleu ou RGB), dont chaque couleur varie entre 0 (noir) et 255 (blanc). Pour contenir les informations d'une image, une classe en C++ nommée *BioImage* fut implémentée. Lors du chargement d'une image, les dimensions de cette image sont enregistrées dans un objet de type *BioImage*, ainsi que la valeur de chacun des pixels de cette image. Ces pixels sont emmagasinés dans un *Array* à une dimension de *unsigned short* (0 à 255), dont la longueur du *Array* équivaut au produit de la largeur et de la hauteur de l'image.

Comme les images sont chargées dans un module en C++ et transférées vers un module en Java (voir figure 4.5), alors le transfert d'images entre ces deux modules via JNI est exécuté comme suit :

1. C++

- Transfert du pointeur d'un objet *image* de type *BioImage* vers JNI.

2. JNI

- Création d'un *Array* pouvant être géré par Java (*jByteArray*) pour contenir des *Array* de *jbyte* (sa longueur étant égale à celle du *Array* de pixels de l'image à transférer). *Peut-être traduit par une liste, pouvant contenir plusieurs listes.*
 - (a) Division du *Array* de pixels de l'objet *image* en *Array* de *jbyte* de longueur N (dans *Biotaxis*, N égale 10 000 bytes).
 - (b) Ajout de la série de *jbyte Array* dans l'objet *jByteArray*.
- Transfert des dimensions $M \times N$ de l'image (soit sa largeur et hauteur en pixels) et de l'objet *jByteArray* vers Java.

3. Java

- Reconstruction de l'image en 2D en divisant le *Array* de *bytes* en plusieurs segments de longueur M à N reprises.

Comme le logiciel *Biotaxis* effectue tous les traitements sur les images dans ses modules en C++, alors le chargement des images en mémoire est réalisé dans ces modules en C++ et seulement les images résultantes d'un traitement sont transférées vers le module en Java. Ce qui permet de minimiser le nombre de transferts requis (aucun transfert d'images en partance de Java vers le C++).

4.2.2 Chargement d'images

Tel que spécifié à la fin de la section précédente, la mise en mémoire des images dans le logiciel *Biotaxis* se fait dans un module codé en C++. Cependant, puisque l'interface utilisateur est implémentée dans le langage Java, il faut donc transférer l'information requise du module en Java vers le module en C++, pour que ce dernier puisse charger l'image en mémoire. Cette information requise est le chemin permettant d'accéder à l'image enregistrée sur le disque dur de l'ordinateur. Un exemple de chemin, sous un système d'exploitation *Windows*, pourrait ressembler à ceci :

« *C : \Biotaxis \Images \Testing \Ciliate \2005_048_163759_147.png* ».

Donc pour charger une image mémoire, le logiciel *Biotaxis* fonctionne de la façon suivante :

1. Java

- Acquisition de la localisation de l'image sur le disque dur (parcours physique).

2. JNI

- Transfert du parcours physique de l'image (du module en Java vers le module en C++).

3. C++

- Valider l'accessibilité de l'image et que son format est bien un type supporté (*png*).
- Charger l'image en mémoire dans un objet de type *BioImage*. Cet objet contient les informations suivantes :
 - Le nom de l'image et le parcours physique vers l'image (« *filepath* »).

- Les dimensions de l'image (largeur et hauteur).
- Un *Array* à une dimension de *unsigned short* contenant le niveau de gris de tous les pixels de l'image.

En résumé, le logiciel *Biotaxis* offre la possibilité de charger une ou des images provenant d'un dossier physique local, dont le format des images doit être de type *png*. Une fois mise en mémoire dans un objet de type *BioImage*, il est alors possible d'effectuer des traitements sur cette image par le biais de fonctionnalités intégrées au module en C++. La section suivante présente les traitements d'images possibles avec le logiciel.

4.2.3 Prétraitement

La section sur le prétraitement, présentée dans le chapitre 3 de l'état de l'art, décrit les méthodes retenues selon les différentes phases de prétraitement. Tel que relaté dans le tableau 3.2, les phases de prétraitement offertes par le logiciel *Biotaxis* sont :

1. Filtrage et détection de contour (optionnel).
 - Aucun filtre
 - Filtre Sobel
 - Filtre Prewitt
2. Seuillage de l'image avec la méthode hystérésis.
3. Tracé de contour de l'élément principal (le phytoplancton) avec la méthode *Convex Hull*.
4. Caractérisation de l'élément principal selon le *Convex Hull*.

L'intégration de chacune de ces méthodes au logiciel *Biotaxis* a requis l'ajout de contrôles (ou bouton) dans l'interface utilisateur en Java, pour permettre la sélection et l'activation de ces méthodes. Puis la conversion de ces méthodes pour permettre leur interprétation par le langage C++. Les points suivants présentent chacune des phases de prétraitement énumérées précédemment.

Filtrage et détection de contour

Les méthodes de détection de contour de Sobel et Feldman [1968] et de Prewitt [1970] ont été implémentées dans l'application *Biotaxis* pour effectuer le filtrage et la détection de

contour d'une image. L'interface utilisateur de l'application offre l'option d'activer ou non cette étape de filtrage et de faire le choix du filtre désiré par le biais d'un menu déroulant.

La sélection d'un filtre occasionne l'insertion de cette étape au début de la séquence de prétraitement de l'image. Cette séquence est ensuite transférée du module en Java vers le module en C++ et déterminera les étapes de prétraitement à effectuer sur les prochaines images à traiter. L'implémentation de la fonction de filtrage et de détection de contour en langage C++ peut se traduire par la portion de code présentée à l'algorithme 1.

Note : La fonction « FiltreDetectionDeContour » (algorithme 1) reçoit en paramètre une image source (src) et la variable (prod) qui est un multiplicateur. Dans le cas du filtre Prewitt, le multiplicateur aura la valeur 1 et 2 pour le filtre Sobel (voir les figures 3.1 et 3.2 comme référence). La fonction renvoie une image vide (null), si l'image source est vide. Autrement, une image destinataire (dst) est créée pour contenir l'image filtrée, qui est ensuite renvoyée par la fonction.

Advenant l'activation du filtrage d'image dans *Biotaxis*, l'image en sortie de la fonction *FiltreDetectionDeContour* sera utilisée à la phase suivante de seuillage de l'image (à défaut d'activer le filtrage, l'image source sera alors utilisée).

Seuillage de l'image

Tel que défini dans l'état de l'art (sous-section 3.1.2), la méthode de seuillage hystérésis a été retenue pour ce projet de recherche. L'état de l'art décrit en détail le fonctionnement de la méthode hystérésis, c'est pourquoi la suite de cette section décrit seulement l'intégration de cette méthode au logiciel *Biotaxis*.

Cette phase de prétraitement utilisant la méthode hystérésis est toujours exécutée avant que la phase de tracé de contour de l'élément principal soit effectuée. Cette méthode est implémentée en C++ et suit la séquence de traitements suivante :

1. Acquérir les paramètres liés à l'histogramme de l'image traitée permettant le calcul des seuils S_b et S_h (voir sous-section 3.1.2).
2. Calculer les valeurs de seuillage S_b et S_h .
3. Produire une image A en noir et blanc selon le seuillage le plus bas (S_b).
4. Produire une image B en noir et blanc selon le seuillage le plus haut (S_h).

Algorithme 1 Filtrage et détection de contour

```
1: function FILTREDETECTIONDECONTOUR(Image src, int prod)
2:   if src.isEmpty() then
3:     return null
4:   end if
5:   int hauteur = src.getHeight()
6:   int largeur = src.getWidth()
7:   Image dst = new Image(hauteur - 2, largeur - 2)
8:   for (int y = 1, y < hauteur - 1, y = y + 1) do
9:     for (int x = 1, x < largeur - 1, x = x + 1) do
10:      int horizontal = -src(y - 1, x - 1) - (prod * src(y - 1, x)) - src(y - 1, x + 1)
11:      horizontal += src(y + 1, x - 1) + (prod * src(y + 1, x)) + src(y + 1, x + 1)
12:      int vertical = -src(y - 1, x - 1) - (prod * src(y, x - 1)) - src(y + 1, x - 1)
13:      vertical += src(y - 1, x + 1) + (prod * src(y, x + 1)) + src(y + 1, x + 1)
14:      int couleur = (abs(horizontal) + abs(vertical))/4
15:      if couleur < 0 then
16:        couleur = 0
17:      else
18:        if couleur > 255 then
19:          couleur = 255
20:        end if
21:      end if
22:      dst(y - 1, x - 1) = 255 - couleur
23:    end for
24:  end for
25:  return dst
26: end function
```

5. Parcourir l'image *B* et lorsqu'un pixel blanc est détecté, vérifier si ce dernier à un pixel blanc mitoyen dans l'image *A* (selon une projection de cette coordonnée dans l'image *A*).
6. Effectuer le transfert de tous les pixels blancs connectés au pixel mitoyen de l'image *B* (principe d'inondation). Voir la figure 3.3 comme référence.

L'image *A* résultante du seuillage par la méthode hystérésis, contient des pixels blancs représentant la surface couverte par l'élément principal (la forme du phytoplancton) et les pixels noirs, l'arrière-plan de cet élément (soit de l'eau, dans le cas des images de phytoplancton).

Tracé de contour de l'élément principal

La phase de tracé de contour de l'élément principal à l'aide de la méthode *Convex Hull*, est une phase intermédiaire entre le seuillage de l'image et la caractérisation de celle-ci. Ainsi, toutes les séquences de traitement de *Biotaxis* requérant la caractérisation de l'image, passeront préalablement par cette phase.

Celle-ci se retrouve également dans un module en C++ et requiert le passage en paramètre d'une image en noir et blanc (soit celle en sortie du seuillage d'image). L'ordre de traitement de cette phase est la suivante :

1. Acquérir les coordonnées des limites cartésiennes de la surface blanche de l'image (soit du pixel blanc le plus à gauche, à droite, en haut et en bas).
2. Découper l'image en une zone rectangulaire minimale pouvant contenir les quatre coordonnées trouvées à l'étape 1.
3. Copier les pixels de cette zone dans une nouvelle image.
4. Tracer le contour de l'élément principal en parcourant les bordures de l'image obtenue à l'étape 3 (voir la sous-section 3.1.3 de l'état de l'art). Lors du tracé de contour :
 - (a) Chacune des coordonnées des coins du contour est conservée dans une liste.
 - (b) Toutes les longueurs des vecteurs formant le tracé de contour sont additionnées pour obtenir le périmètre du contour, ainsi que la sommation de l'angle formé par chacun de ces vecteurs.
5. Tracer la plus grande diagonale possible pouvant être contenue à l'intérieur du contour (la diagonale maximale). Cette diagonale est obtenue grâce au calcul de la distance entre chacune des coordonnées contenues dans la liste de coins du contour (étape 4a). La plus grande distante représentant la diagonale maximale.
6. Calculer l'angle de la diagonale maximal avec le plan horizontal.
7. Effectuer la rotation de l'image selon l'angle trouvé à l'étape 6 afin de positionner la diagonale sur un plan horizontal.

La suite des traitements appliqués à l'image, relève de la phase suivante de caractérisation.

Caractérisation de l'élément principal

Cette phase du prétraitement permet d'identifier différents paramètres propres à l'élément principal d'une image. La sous-section 3.1.4 de l'état de l'art contient un tableau présentant les 20 caractéristiques ayant été utilisées pour cette recherche (voir tableau 3.3).

L'analyse de ce tableau permet d'identifier les variables nécessaires au calcul de l'ensemble des caractéristiques de celui-ci. Ces variables sont : D_{max} , X_{max1} , X_{max2} , A_{c1} , A_{c2} , A_p , P_c , C_{mean} .

- La valeur de D_{max} , qui représente la longueur de la plus grande diagonale, est obtenue lors de l'étape 5 de la phase de tracé de contour.
- Les variables X_{max1} et X_{max2} représentent respectivement la distance minimale et maximale entre la diagonale (sur un plan horizontal) et les limites inférieures et supérieures du tracé de contour. La plus petite distance étant assignée à X_{max1} et la plus grande à X_{max2} .
- A_{c1} et A_{c2} définissent l'aire minimale et maximale du *Convex Hull* (soit l'aire du *Convex Hull* divisé en deux par la diagonale maximale). Ces valeurs sont acquises en effectuant la sommation de la distance entre la diagonale maximale avec la courbe supérieure et inférieure à celle-ci (se référer à la figure 3.8).
- La variable A_p qui est l'aire de l'élément principal équivaut au nombre de pixels blancs contenus dans le *Convex Hull*.
- P_c est le périmètre du *Convex Hull* ayant été calculé à l'étape 4b de la phase de tracé de contour.
- Finalement, la variable C_{mean} , qui est la moyenne des courbes du *Convex Hull*, est obtenue par la division de la sommation des angles, calculée à l'étape 4b de la phase de tracé de contour, divisé par le nombre d'angles.

Une fois que la valeur de chacune des variables ci-haut mentionnées est établie, il suffit d'effectuer les calculs énoncés dans le tableau 3.3 pour obtenir toutes les caractéristiques propres à l'élément principal de l'image traitée.

4.2.4 Discrimination

La discrimination de catégories sert à éliminer, parmi les catégories possibles (21 pour le projet *Biotaxis*), celles ayant le plus de différences avec l'image traitée. Tel que décrit à la section 3.2 de l'état de l'art, c'est l'arbre décisionnel à intervalles qui a été choisi comme méthode pour effectuer la discrimination de catégories. La génération d'un arbre pour chacune des 20 caractéristiques a été intégrée au module en Java dans *Biotaxis*.

Normalisation des limites entre -1 et 1

La génération de chacun des arbres requiert d'abord l'obtention de la limite inférieure et supérieure d'une même caractéristique pour l'ensemble des catégories d'un groupe de classification (la sous-section 3.2.2 décrit comment ces valeurs sont calculées). Ensuite, l'ensemble des limites doit être normalisé afin de pouvoir être contenu entre les intervalles de -1 et 1. Cette étape est réalisée grâce aux calculs suivants :

$$div = \left\| \frac{max - min}{2} \right\| \quad (4.1)$$

$$s = -(div + min) \quad (4.2)$$

Où *div* = *Divider*, *s* = *Shift*, *min* = *Minimum Value* et *max* = *Maximum Value*.

Note : la variable min représente la valeur minimale de l'ensemble des limites inférieures d'une caractéristique parmi les catégories du groupe de classification. La variable max est la valeur maximale de l'ensemble des limites supérieures.

Lorsque les variables *div* et *s* sont calculées, chacune des limites de la caractéristique traitée doit être normalisée. Cette opération est réalisée à l'aide de la formule suivante :

$$sv = \frac{lv + s}{div} \quad (4.3)$$

Où *div* = *Divider*, *s* = *Shift*, *sv* = *Scaled Value* et *lv* = *Limit Value*.

Création d'un arbre décisionnel à intervalles

Suite à la normalisation des limites, les méthodes menant à la génération d'un arbre décisionnel à intervalles sont appelées par le logiciel. La figure 3.9 de l'état de l'art décrit parfaitement les premières étapes nécessaires à la génération d'un tel arbre. Cependant, la figure 3.10 (après la figure 3.9) requiert de plus amples détails sur le processus utilisé pour pouvoir créer l'arbre décisionnel à intervalles présenté dans cette figure.

1. Une liste est créée pour contenir tous les noeuds de l'arbre (un noeud pour chacune des catégories du groupe de classification). Chaque noeud est un objet contenant les informations suivantes (voir sous-section 3.2.4) :

- Le nom de la catégorie d'appartenance.
- L'intervalle minimal et maximal de la caractéristique traitée.

- La valeur minimale et maximale des enfants du noeud.
 - Un pointeur vers le noeud parent.
 - Un pointeur vers l'enfant de gauche du noeud et un autre vers l'enfant de droite.
2. Suite au remplissage de la liste de noeuds, celle-ci est alors triée en fonction de la limite inférieure des intervalles de chacun des noeuds de la liste.
 3. L'étape suivante consiste à transmettre cette liste nouvellement triée à la fonction de création de l'arbre.

Les algorithmes présentés ci-après sont des exemples de fonctions provenant du projet *Biotaxis* permettant la création d'un arbre décisionnel à intervalles.

Lorsqu'un arbre décisionnel est créé, il est alors possible d'effectuer la discrimination de catégories en appliquant les étapes décrites à titre d'exemple à la sous-section 3.2.4 de l'état de l'art.

4.2.5 Classification SVM

Cette étape d'attribution d'une classe à une image de phytoplancton est basée sur la méthode de classification SVM. L'implémentation de ce processus fut simplifiée grâce à l'usage de la librairie *jlibsvm* qui est implémentée en Java [Soergel, 2012] et dérivée de la librairie *libsvm* ayant été citée dans les articles de Alvarez *et al.* [2012] et Sosik et Olson [2007]. Puisque les auteurs de ces articles ont obtenu une bonne précision de la classification grâce à son usage (voir tableau 2.1), il fut donc approprié d'utiliser cette librairie dans *Biotaxis* pour effectuer la classification d'images.

La section « Documentation » du site internet décrivant la librairie *jlibsvm* [Soergel, 2012], énumère les étapes nécessaires à la génération d'un modèle SVM avec cette librairie. Voici la traduction en français des étapes citées sur ce site écrit en anglais :

1. Initialisation de l'objet « *KernelFunction* ».
2. Configuration de l'objet de paramétrisation du classificateur SVM « *SvmParameter* » (définit le comportement et le type de traitement du classificateur).
3. Initialisation de la sous-classe *SvmProblem* et instanciation de la liste d'objet d'entraînement (chaque *problem* est un objet contenant la catégorie et les caractéristiques d'une image d'entraînement).
4. Initialisation de la sous-classe SVM.

Algorithme 2 Création d'un arbre décisionnel à intervalles (fonction 1)

```
1: function BUILDTREE(List < Node > treeNodes)
2:   if treeNodes.isEmpty() then
3:     return
4:   end if
5:   int pivot = (treeNodes.size() >>> 1) + 1
6:   Node root = treeNodes.get(pivot - 1)
7:   setChildNodes(treeNodes, root, 1, pivot - 1, -1) //Set Left Side
8:   setChildNodes(treeNodes, root, pivot+1, treeNodes.size(), 1) //Set Right Side
9:   setTreeNodeMinMax(root)
10: end function
```

Algorithme 3 Création d'un arbre décisionnel à intervalles (fonction 2)

```
1: function SETCHILDNODES(List < Node > treeNodes, Node parent, int low,
   int high, int side)
2:   if low < 1 || high > treeNodes.size() || low > high then
3:     return
4:   end if
5:   if low == high then
6:     Node node = treeNodes.get(low - 1)
7:     node.setParent(parent)
8:     if side < 0 then
9:       parent.setLeftChild(node)
10:    else
11:      parent.setRightChild(node)
12:    end if
13:    return
14:  end if
15:  int pivot = ((high - (low - 1)) >>> 1) + 1 + (low - 1)
16:  Node node = treeNodes.get(pivot - 1)
17:  node.setParent(parent)
18:  if side < 0 then
19:    parent.setLeftChild(node)
20:  else
21:    parent.setRightChild(node)
22:  end if
23:  if pivot - 1 ≥ low then
24:    setChildNodes(treeNodes, node, low, pivot - 1, -1) //Set Left Side
25:  end if
26:  if pivot + 1 ≤ high then
27:    setChildNodes(treeNodes, node, pivot + 1, high, 1) //Set Right Side
28:  end if
29: end function
```

Algorithme 4 Création d'un arbre décisionnel à intervalles (fonction 3)

```
1: function SETTREENODESMINMAX(Node node)
2:   if node.getLeft() != null then
3:     setTreeNodesMinMax(node.getLeftChild())
4:   end if
5:   if node.getRight() != null then
6:     setTreeNodesMinMax(node.getRightChild())
7:   end if
8:   if node.getParent() != null then
9:     Node parent = node.getParent()
10:    parent.setMinLimit(node.getMinLimit())
11:    parent.setMaxLimit(node.getMaxLimit())
12:   end if
13: end function
```

5. Appeler la fonction *SVM.train(problem)* pour générer un modèle SVM (le modèle est utilisé pour faire des prédictions de classification).

À partir de cette liste d'étapes, l'intégration de la librairie *jlibsvm* au logiciel *Biotaxis* fut réalisée de la façon suivante :

1. Ajout de la librairie *jlibsvm* au projet en Java de *Biotaxis*.
2. Création d'une classe de classification SVM en Java. Cette classe servant d'intermédiaire entre le projet *Biotaxis* et la librairie *jlibsvm*.
3. La liste des objets de la classe SVM et leurs configurations pour ce projet est la suivante :

- (a) Instanciation de l'objet *kernelRBF* de type « *GaussianRBFKernel* ».
- (b) Instanciation de l'objet *param* de type « *SvmParameter* » et configuration des paramètres de l'objet.

```
param.C = 1f;           // Default. Set the parameter C of C-SVC
param.p = 0.1f;         // Default. Set epsilon in loss function
param.nu = 0.5f;        // Default. Set the parameter nu of nu-SVC
param.eps = 1e-3f;      // Default. Set tolerance of termination.
// Set cache memory size in MB
param.cache_size = 40;
// Whether to use the shrinking heuristics
param.shrinking = true; // Default
// Whether to train a SVC or SVR model
```

```

param.probability = false;
// Set type of kernel function
param.kernel = kernelRBF;
// Set false One Vs All
param.oneVsAllMode = OneVsAllMode.None;
// Set true One Vs One
param.allVsAllMode = AllVsAllMode.AllVsAll;

```

- (c) Instanciation de l'objet *problem* de type
« *MultiClassProblem* < *String*, *SparseVector* > ».
- i. Ajout des données d'entraînement (soit 20 caractéristiques par image d'entraînement, donc 20 données par image)
 - Type « *String* » = nom de la catégorie d'appartenance de l'image.
 - Type « *SparseVector* » = liste d'objets comprenant chacun l'identifiant et la valeur pour chacune des 20 caractéristiques de l'image.
- (d) Instanciation de l'objet *svm* de type « *MultiClassificationSVM* ».
- (e) Instanciation de l'objet *svmModel* de type « *MultiClassModel* », où *svmModel* = *svm.train(problem, param)*.
- 4. Au terme de la création de l'objet *svmModel*, celui-ci peut être utilisé pour faire des prédictions de classification. Le processus de prédiction de la catégorie d'appartenance d'une image est effectué comme suit :
 - (a) Création d'un objet **vx** de type « *SparseVector* » à partir des 20 caractéristiques de l'image à classer.
 - (b) Prédiction de la catégorie d'appartenance de l'image, où le nom de la catégorie = *svmModel.predictLabel(vx)*.

Note : la configuration de la classe SVM présentée ci-haut permet de faire la classification de multi classes en utilisant l'approche un contre un (« One – vs – One »).

4.2.6 Précision de la classification et temps de traitement

Le calcul de la précision de la classification est une évaluation de la qualité du processus d'attribution d'une catégorie à une image donnée. Puisque les images utilisées sont déjà classifiées (voir la sous-section 4.1.1), il est donc possible d'utiliser ces informations pour

calculer la précision de la classification et donc évaluer sa qualité. Voici un pseudocode présentant la logique de calcul de la moyenne de la précision :

Algorithme 5 Calcul de la moyenne de précision

```

1: procedure CALCULMOYENNEDEPRECISION(categorieObtenues, categorieReelles)
2:   MoyenneDePrécision  $\leftarrow$  0.0
3:   SommeBonneClassification  $\leftarrow$  0.0
4:   for  $i \leftarrow 0, TotalImagesTest$  do
5:     if categorieObtenues[ $i$ ] == categorieReelles[ $i$ ] then
6:       SommeBonneClassification  $\leftarrow$  SommeBonneClassification + 1.0
7:     end if
8:   end for
9:   MoyenneDePrécision  $\leftarrow$  SommeBonneClassification / TotalImagesTest
10:  return MoyenneDePrécision
11: end procedure

```

En résumé, cette fonction fait une comparaison entre la catégorie attribué à une image après sa classification (*categorieObtenues*) versus sa véritable catégorie d'appartenance (*categorieReelles*). Donc si les deux valeurs sont identiques, on ajoute 1 au nombre de bonnes classifications. Après avoir fait cette validation pour l'ensemble des images testées, on divise le nombre de bonnes classifications par le nombre total d'images testées pour obtenir la moyenne de précision de la classification globale. En apportant quelques modifications à ce pseudocode, il est également possible d'évaluer la précision de la classification obtenue pour chacune des catégories du groupe de classification.

Pour ce qui est du temps de traitement d'une image, qui est représenté par l'intervalle de temps requis pour passer du chargement de l'image jusqu'à l'attribution d'une catégorie. Ce dernier est obtenu par le calcul de la différence entre le temps de départ X (avant le chargement de l'image) et le temps final Y (après l'attribution d'une catégorie). La section 3.4 de l'état de l'art présente de quelle façon la temps de traitement en millisecondes est calculé en Java.

Une fois que tous les temps de traitement d'images sont compilés, il est possible d'obtenir le temps de traitement moyen global de l'ensemble du processus par le biais de l'équation suivante :

$$ttmg = (\sum_{i=0}^N tr[i]) / N \quad (4.4)$$

Où $ttmg$ = Temps de Traitement Moyen Global, N = Nombre total d'images et tr = Temps Requis.

Cette formule pouvant être appliquée pour chaque catégorie et pour l'ensemble des catégories testées (représentant la moyenne du temps de traitement global).

4.2.7 Description des options de traitement du logiciel *Biotaxis*

Cette section présente les trois options de traitement offertes par le logiciel *Biotaxis*. Ces options, accessibles par le biais d'onglet, sont décrites ci-après.

Onglet « Analysis »

D'abord, le logiciel *Biotaxis* offre la possibilité de faire une analyse du prétraitement effectué sur une image. Accessible via l'onglet « Analysis », cette fenêtre permet de consulter visuellement le rendu de chacune des phases de prétraitement effectuées sur une image. La figure 4.6 illustre le contenu de cette fenêtre.

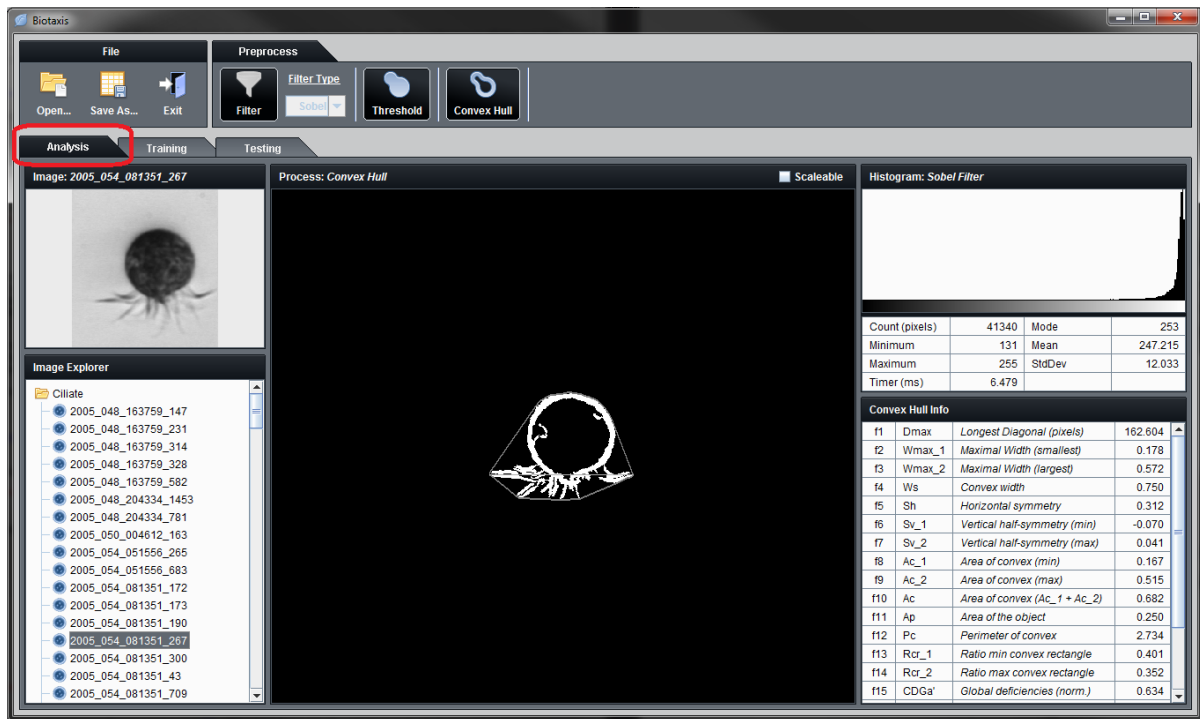


Figure 4.6 Interface après la sélection de l'onglet « Analysis » (logiciel *Biotaxis*)

En bref, les fonctionnalités illustrées à la figure 4.6 de la fenêtre « Analysis » se résument comme suit :

1. Option « File » du menu en en-tête
 - (a) Bouton « Open... » : Ouvre une fenêtre de navigation permettant de charger une ou des image(s) de format *png* dans l'application.

- (b) Bouton « *Save As...* » : Permet de sauvegarder en format *csv* (« *Comma-Separated Values* ») les valeurs de caractérisation de l'image traitée.
- (c) Bouton « *Exit* » : Ferme l'application.

2. Option « *Preprocess* » du menu en entête

- (a) Bouton « *Filter* » : Si le bouton n'est pas sélectionné, alors aucun filtre ne sera appliqué à l'image lors du prétraitement, l'inverse si il est activé.
- (b) Menu déroulant « *Filter Type* » : Lorsque le bouton « *Filter* » est activé, il est alors possible de sélectionner la méthode de filtrage de l'image. Les options de filtrage étant les méthodes de Sobel et Prewitt.
- (c) Bouton « *Threshlod* » : Ce bouton est toujours activé et le passage du curseur de la souris sur celui-ci permet de visualiser l'application de la méthode hystérésis sur l'image traitée.
- (d) Bouton « *Convex Hull* » : La sélection de ce bouton permet d'exécuter la phase complète de prétraitement d'une image (voir le tableau 3.2 pour la liste de traitement). Le rendu visuel de cette sélection étant l'affichage de l'image traitée après l'application de la méthode *Convex Hull* et les valeurs de caractérisation de l'image.

3. Onglet « *Analysis* »

- (a) Fenêtre « *Images* » : Affiche l'image originale, soit sans traitement, ainsi que le nom du fichier image.
- (b) Fenêtre « *Image Explorer* » : Affiche la liste des images et selon le cas, les dossiers contenant les images chargées dans le logiciel (voir bouton « *Open...* »).
- (c) Fenêtre « *Process* » : Affiche le rendu visuel de l'image traitée suite à l'application d'un filtre, d'un seuillage (« *threshold* ») ou du *Convex Hull*. L'option « *Scaleable* » permet d'afficher l'image dans cette fenêtre en mode « plein écran ».
- (d) Fenêtre « *Histogram* » : Affiche l'histogramme de l'image et les informations liées à l'analyse de l'histogramme (nombre de pixels, niveau de gris minimal et maximal, mode, moyenne et écart-type).
- (e) Fenêtre « *Biotaxis info* » : Affiche les valeurs de caractérisation de l'image (après la sélection du bouton « *Biotaxis* »).

Onglet « Training »

La fenêtre d'entraînement, accessible par la sélection de l'onglet « *Training* », permet la gestion et l'entraînement de groupes de classification (voir la figure 4.7).

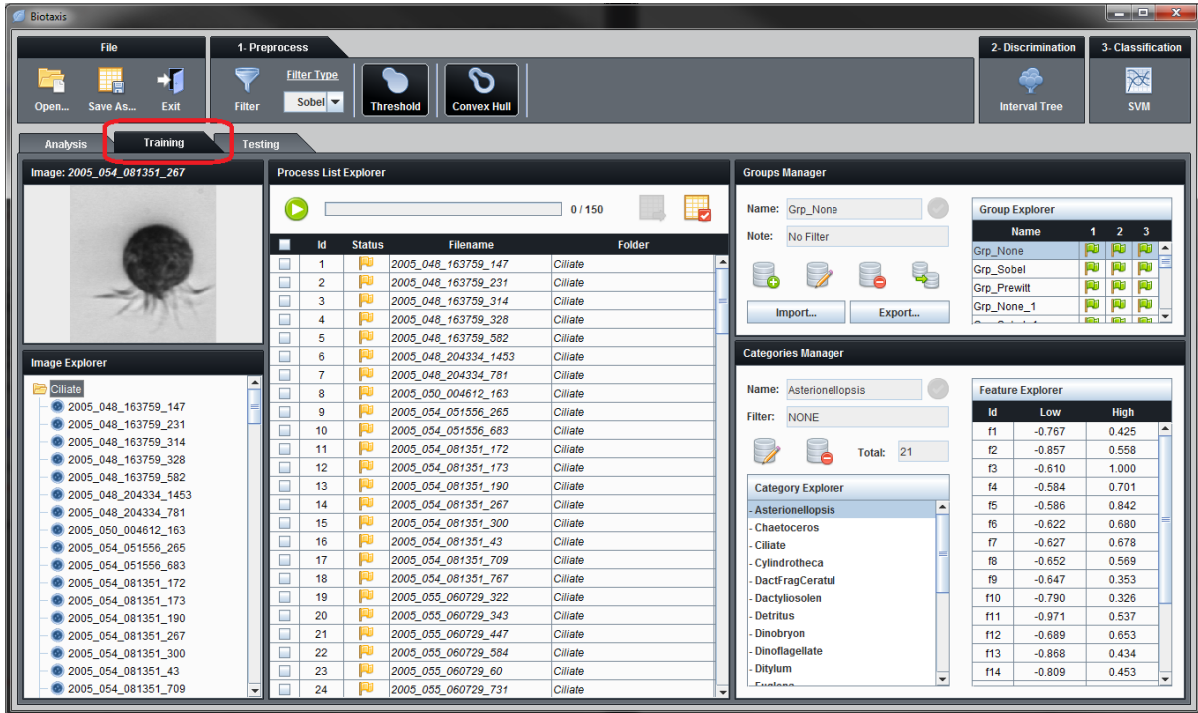


Figure 4.7 Interface après la sélection de l'onglet « *Training* » (logiciel *Biotaxis*)

Cette fenêtre offre les fonctionnalités nécessaires à la création d'un groupe de classification. Ce groupe peut ensuite être utilisé pour effectuer des tests de classification sur des images (voir la description suivante *Onglet « Testing »*). Voici la description des composants de cette fenêtre :

1. Option « *File* » du menu en en-tête (voir les descriptions 1a, 1b et 1c de l'onglet « *Analysis* »).
2. Option « *1 - Preprocess* » du menu en entête (voir les descriptions 2a à 2d de l'onglet « *Analysis* »).
3. Option « *2 - Discrimination* » du menu en en-tête.
 - (a) Bouton « *Interval Tree* » : Actif après la phase de prétraitement d'un groupe de classification. Permet la génération des arbres décisionnels à intervalles du groupe sélectionné (voir sous-section 4.2.4).

4. Option « 3 - *Classification* » du menu en en-tête.

- (a) Bouton « *SVM* » : Actif après la phase de prétraitement d'un groupe de classification. Permet la génération d'un modèle SVM global du groupe sélectionné (voir sous-section 4.2.5).

5. Onglet « *Training* »

- (a) Fenêtre « *Images* » (voir la description 3a l'onglet « *Analysis* »).
- (b) Fenêtre « *Image Explorer* » (voir les descriptions 3b de l'onglet « *Analysis* »).
- (c) Fenêtre « *Process List Explorer* »
 - i. Bouton « *Start* » : Exécute le prétraitement de liste d'images.
 - ii. Barre de défilement et compteur : Affiche la progression de la phase de prétraitement.
 - iii. Bouton « *Transfer* » : Fait le transfert de l'ensemble des caractérisations d'images d'une catégorie vers un groupe de classification.
 - iv. Bouton « *Delete* » : Efface les images de la liste ayant été sélectionnées (avec un crochet).
 - v. Liste d'images à traiter : Chaque ligne de la liste présente : une boîte de sélection ; un identifiant ; le statut du traitement (en attente, complété et échoué) ; le nom du fichier image ; le nom du dossier de l'image.
- (d) Fenêtre « *Group Manager* » : Permet la gestion des groupes de classification.
 - i. Champs texte : Affichage et édition du nom et de la description (« *Tooltip* ») d'un groupe de classification.
 - ii. Bouton « *Add* » : Permet la création d'un groupe de classification.
 - iii. Bouton « *Edit* » : Rend les champs texte « *Name* » et « *Note* » éditables et active le bouton de confirmation (bouton avec un crochet).
 - iv. Bouton « *Delete* » : Fait la suppression du groupe de classification sélectionné.
 - v. Bouton « *Duplicate* » : Crée une copie de l'ensemble des informations du groupe sélectionné et génère un nouveau groupe.

- vi. Bouton « *Import...* » : Permet l'importation des informations d'un groupe de classification provenant d'un disque physique (fichier de format **.bio*) et ajout du groupe à la fenêtre « *Group Explorer* ».
 - vii. Bouton « *Export...* » : Permet l'exportation des informations d'un groupe de classification sur un disque physique (crée un fichier de format **.bio*).
 - viii. Fenêtre « *Group Explorer* » : Affiche la liste de groupes de classification disponible. Les drapeaux identifient l'exécution ou non des différentes phases de traitement (1 - Prétraitement, 2 - Discrimination, 3- Classification).
- (e) Fenêtre « *Categories Manager* » : Permet la gestion des catégories appartenant au groupe de classification sélectionné.
- i. Champs texte : Affichage et édition du nom d'une catégorie. Affichage du filtre et du nombre total de catégories définies par le groupe de classification sélectionné.
 - ii. Bouton « *Edit* » : Rend le champ texte « Name » éditable et active le bouton de confirmation (bouton avec un crochet).
 - iii. Bouton « *Delete* » : Fait la suppression de la catégorie sélectionnée.
 - iv. Fenêtre « *Category Explorer* » : Liste des catégories définies par le groupe sélectionné. La sélection d'une catégorie dans la liste permet l'affichage des informations liées à cette catégorie.
 - v. Fenêtre « *Feature Explorer* » : Liste de la valeur minimale et maximale de chacune des caractéristiques de la catégorie sélectionnée.

Onglet « *Testing* »

Pour effectuer des tests de classification, l'onglet « *Testing* » doit-être sélectionné. Cette fenêtre offre la possibilité d'effectuer des tests de classification d'image, de donner la précision de la classification (si l'image est déjà préclassifiée) et de prédire la catégorie d'appartenance d'une image (étant valable **si** la catégorie d'appartenance de l'image testée est définie par le groupe de classification utilisé). La figure 4.8 illustre les fonctionnalités disponibles lors de la sélection de l'onglet « *Testing* ».

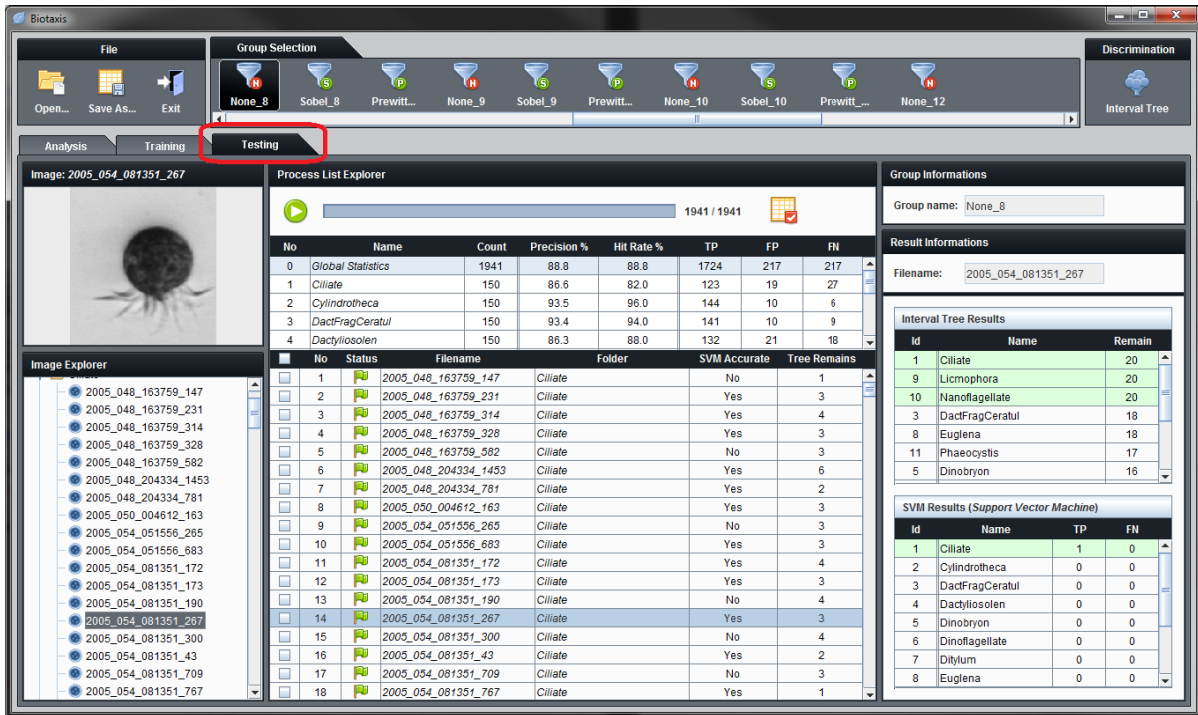


Figure 4.8 Interface après la sélection de l'onglet « *Testing* » (logiciel *Biotaxis*)

La description des options de cette fenêtre est énoncée ci-après :

1. Option « *File* » du menu en en-tête (voir les descriptions 1a, 1b et 1c de l'onglet « *Analysis* »).
2. Option « *Group Selection* » du menu en entête : présente une liste de bouton permettant de sélectionner un groupe de classification qui sera utilisé pour la phase de test de classification d'image(s).
3. Option « *Discrimination* » du menu en en-tête : permet d'utiliser la solution 1 (avec discrimination), si le bouton est activé, ou sinon, la solution 2 (sans discrimination). Le chapitre 3 de l'état de l'art présente une description de ces deux solutions.
4. Onglet « *Testing* »
 - (a) Fenêtre « *Images* » (voir la description 3a l'onglet « *Analysis* »).
 - (b) Fenêtre « *Image Explorer* » (voir la descriptions 3b de l'onglet « *Analysis* »).
 - (c) Fenêtre « *Process List Explorer* ».
 - i. Bouton « *Start* » : Exécute la classification de liste d'images.
 - ii. Barre de défilement et compteur : Affiche la progression de la classification.

- iii. Bouton « *Transfer* » : Fait le transfert de l'ensemble des caractérisations d'images d'une catégorie vers un groupe de classification.
- iv. Bouton « *Delete* » : Efface les images de la liste ayant été sélectionnées (avec un crochet).
- v. Liste de compilation des résultats de test : Chaque ligne de la liste présente : un identifiant ; un nom de catégorie ; le nombre d'images préclassifiées⁴ appartenant à cette catégorie ; la précision de la classification (%) ; le ratio de bonne classification (« *Hit Rate* ») en % ; le compte de valeurs vraies positives (TP) ; le compte de valeurs fausses positives (FP) ; le compte de valeurs fausses négatives (FN). Voir la section 3.4 pour plus de détails.
- vi. Liste d'images à traiter : Chaque ligne de la liste présente : une boîte de sélection ; un identifiant ; le statut du traitement (en attente, complétée et échouée) ; le nom du fichier image ; le nom du dossier de l'image ; si l'image est préclassifiée ; le nombre de catégories restantes (après discrimination).

(d) Fenêtre « *Group Informations* ».

- i. Champ texte « *Group name* » : Nom du groupe de classification sélectionné en en-tête.

(e) Fenêtre « *Result Informations* ».

- i. Champ texte « *Filename* » : Nom de l'item sélectionné dans la fenêtre « Process List Explorer ».
- ii. Fenêtre « *Interval Tree Results* » : Résultats de la discrimination de l'image par le passage dans les arbres décisionnels à intervalles. Chaque ligne de la liste donne comme information : un identifiant de catégorie ; le nom de la catégorie ; le nombre d'arbre où cette catégorie n'a pas été discriminée.
- iii. Fenêtre « *SVM Results* » : Présente la résultat de la classification SVM d'une image (l'image doit être préclassifiée pour obtenir des valeurs). Chaque ligne de la liste présente : un identifiant de catégorie ; le nom de la catégorie ; la valeur vraie positive (TP), s'il y a lieu ; la valeur fausse négative (FN), s'il y a lieu.

⁴Pour déterminer si une image est préclassifiée, le nom de l'image et le nom du dossier contenant l'image sont comparés avec la liste de catégories définie par le groupe de classification sélectionné. Si l'un des deux nom est présent dans la liste, alors l'image sera considérée comme préclassifiée et ayant pour catégorie d'appartenance le nom du fichier ou dossier concordant avec la liste.

4.3 Tests des différentes phases de classification

Afin de valider l'exactitude des résultats obtenus lors de la classification d'images avec le logiciel *Biotaxis*, différents tests ont été effectués en cours de développement. D'abord, en ce qui a trait à la qualité du code, l'approche de développement par tests a été utilisée pour le développement du logiciel *Biotaxis*. Ainsi, l'ajout de fonctions, méthodes et classes au projet *Biotaxis* a amené en contrepartie, l'ajout de tests de résultats, l'analyse continue du résultat des traces (« *Logger* ») et la validation des métriques de qualité du code [Metrics, 2014]. Cette examen des métriques démontre que la programmation du logiciel a été faite dans le respect des indicateurs de qualité⁵. *Il est possible de consulter un exemplaire de l'analyse des métriques du code en Java de Biotaxis à la figure C.1 de l'annexe C.*

En plus des tests de vérification du code ci-haut mentionnés, le rendu de chacune des étapes de classification a également fait l'objet de tests pour valider l'exactitude des informations renvoyées. Pour débiter, les différents traitements possibles de la phase de prétraitement ont pu être testés à l'aide de la fenêtre d'analyse du prétraitement (sous-section 4.2.7 - voir la description de l'onglet « *Analysis* »). Le fait de pouvoir visualiser le rendu de chacune des étapes de prétraitement, ainsi que consulter les valeurs de caractérisation de l'image ont permis de faciliter la détection de problème et d'en accélérer la correction.

Pour la phase de discrimination, l'usage de trace et l'impression à la console des arbres décisionnels par intervalles ont permis de valider la conformité des arbres générés. Un exemple de génération d'informations en console dans le logiciel *Eclipse* est présenté à la figure 4.9.

Finalement, les tests liés à la classification d'images avec la méthode SVM ont été réalisés par la consultation des résultats de la précision de la classification à l'aide de la fenêtre de tests de *Biotaxis* (sous-section 4.2.7 - voir la description de l'onglet « *Testing* »). Ayant utilisé une librairie externe, soit *jlibsvm*, pour effectuer les fonctionnalités de la méthode SVM, il fut donc supposé que cette librairie ait déjà fait l'objet de tests par l'auteur de celle-ci [Soergel, 2012].

⁵Les indicateurs de qualité d'une métrique sont définis sous la forme d'intervalles, dont la valeur de la métrique doit respecter pour être identifiée comme étant de qualité. Par exemple, l'analyse des métriques du code en Java avec l'outil Metrics [2014], présente en bleu les métriques qui respectent les indicateurs de qualité et en rouge, ceux qui ne les respectent pas (voir la figure C.1 de l'annexe C pour un exemple d'analyse des métriques).

```

<terminated> Main [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (May 19, 2014, 7:57:25 PM)

TREE NODES (0) :
0. ( CatId: 2 [-0.639, -0.273] Min: -1.000, Max: 1.000)
1. ( CatId: 3 [-0.737, 0.170] Min: -1.000, Max: 1.000) ( CatId: 6 [-0.607, -0.479] Min: -0.635, Max: 0.372)
2. ( CatId: 0 [-0.836, 0.403] Min: -1.000, Max: 0.937) ( CatId: 11 [-0.691, -0.437] Min: -0.734, Max: 1.000) ( CatId:
3. ( CatId: 13 [-0.914, 0.799] Min: -1.000, Max: 0.937) ( CatId: 12 [-0.768, -0.133] Min: -0.768, Max: -0.133) ( CatId:
4. ( CatId: 9 [-1.000, 0.937] Min: -1.000, Max: 0.937)
TREE NODES (1) :
0. ( CatId: 2 [-0.641, -0.269] Min: -1.000, Max: 1.000)
1. ( CatId: 12 [-0.683, -0.072] Min: -1.000, Max: 1.000) ( CatId: 8 [-0.533, 0.445] Min: -0.624, Max: 0.868)
2. ( CatId: 4 [-0.816, -0.156] Min: -1.000, Max: 0.491) ( CatId: 5 [-0.645, 0.352] Min: -0.667, Max: 1.000) ( CatId:
3. ( CatId: 9 [-0.898, 0.012] Min: -1.000, Max: 0.491) ( CatId: 3 [-0.784, 0.340] Min: -0.784, Max: 0.340) ( CatId: 1
4. ( CatId: 0 [-1.000, 0.491] Min: -1.000, Max: 0.491)
TREE NODES (2) :
0. ( CatId: 2 [-0.672, -0.218] Min: -1.000, Max: 1.000)
1. ( CatId: 3 [-0.764, 0.375] Min: -1.000, Max: 0.915) ( CatId: 8 [-0.374, 0.445] Min: -0.610, Max: 1.000)
2. ( CatId: 4 [-0.793, -0.069] Min: -1.000, Max: 0.915) ( CatId: 14 [-0.725, -0.012] Min: -0.756, Max: -0.012) ( CatId:
3. ( CatId: 9 [-0.913, 0.017] Min: -1.000, Max: 0.915) ( CatId: 12 [-0.782, 0.105] Min: -0.782, Max: 0.105) ( CatId:
4. ( CatId: 0 [-1.000, 0.915] Min: -1.000, Max: 0.915)

```

Figure 4.9 Exemple d'informations produites en console lors de la génération d'arbres décisionnels à intervalles (logiciel *Eclipse*)

4.4 Méthode de classification d'image avec le logiciel *Biotaxis*

Cette section traite de la séquence d'opérations requise pour faire la classification d'une image à l'aide du logiciel *Biotaxis*. Le but étant de faire le lien entre le logiciel et l'ensemble des étapes de traitement présentées dans le chapitre 3 « État de l'art » et dans le chapitre 5 « Résultats et Analyse ».

Séquence menant à la classification d'une image avec *Biotaxis*

1. D'abord, avoir deux groupes d'images préclassifiés comprenant un minimum de deux catégories distinct.
2. Entraîner un groupe de classification (se référer à la sous-section 4.2.7 - Onglet « *Training* » pour plus de détails)
 - (a) Sélectionner l'onglet « *Training* ».
 - (b) Créer un groupe de classification.
 - (c) Charger le groupe d'images d'entraînement.
 - (d) Déplacer l'ensemble des images, d'une seule catégorie, de la fenêtre « *Image Explorer* » vers la fenêtre « *Process List Explorer* ».
 - (e) Sélectionner les options de prétraitement désirées (menu en-tête - panneau « *1 - Preprocess* »).

- (f) Exécuter l'entraînement de la catégorie.
 - (g) Transférer le résultat de l'entraînement de la catégorie vers le groupe de classification en appuyant sur le bouton « *Transfer to bioGroup* ».
 - (h) *Répéter à partir de l'étape 4, jusqu'à ce que l'ensemble des catégories soit entraîné.*
 - (i) Appuyer sur le bouton « *Tree Interval* », dans le menu d'en-tête, pour générer les arbres décisionnels à intervalles (utilisé par la solution 1, soit la classification avec discrimination).
 - (j) Appuyer sur le bouton « *SVM* », dans le menu d'en-tête, pour créer le modèle SVM global du groupe et permettre la prédiction de catégorie d'une image.
 - (k) Le groupe de classification nouvellement créé peut maintenant être utilisé pour tester la classification d'images.
3. Tester le groupe de classification sur un ensemble d'images test (se référer à la sous-section 4.2.7 - Onglet « *Testing* » pour plus de détails)
- (a) Sélectionner l'onglet « *Testing* ».
 - (b) Sélectionner le groupe de classification.
 - (c) Charger le groupe d'images test.
 - (d) Déplacer l'ensemble des images test, ou seulement une partie, de la fenêtre « *Image Explorer* » vers la fenêtre « *Process List Explorer* ».
 - (e) **Optionnel** - Activer le bouton « *Tree Interval* » pour effectuer un test de classification selon la solution 1, soit avec discrimination (la solution 2 étant la solution par défaut).
 - (f) Exécuter le test de classification du groupe d'images.
 - (g) Vérifier la précision de la classification du groupe. *Selon les options de pré-traitement choisies, il est possible que la précision de la classification varie. Il est donc souhaitable de créer et tester différents groupes de classification avec des étapes de pré-traitement différentes, afin de déterminer quel est le groupe de classification le plus précis pour les catégories d'images à classifier.*

4.5 Bilan du développement logiciel

L'outil logiciel qui a été réalisé et dont les étapes de développement ont été présentées dans ce chapitre, représente une plateforme de test et de validation des concepts et principes qui ont été à l'étude. L'analyse du développement logiciel permet de faire le lien entre l'étude des phases de traitement énoncée dans le chapitre 3 de l'état de l'art et de leur intégration au logiciel. Ces phases de traitement sont ensuite utilisées par les modules du logiciel pour effectuer : l'analyse du prétraitement d'image ; l'entraînement de groupes de classification ; puis les tests de classification des groupes (voir sous-section 4.2.7).

Dans le chapitre en cours, il y a également une section sur les test qui ont été effectués pour vérifier la qualité du logiciel. Ces tests ont démontré que le logiciel *Biotaxis* respecte les indicateurs de qualité (section 4.3). Finalement, le chapitre termine par une présentation de la séquence d'opérations requise pour effectuer une classification d'image avec le logiciel. Ce chapitre a donc permis d'énoncer les étapes de développement qui ont guidé la réalisation du logiciel et ainsi, permettre l'atteinte de l'objectif sous-jacent. Le chapitre suivant présente les résultats de la classification d'image obtenus avec le logiciel, ainsi qu'une analyse de ces résultats.

CHAPITRE 5

RÉSULTATS ET ANALYSE

5.1 Présentation des résultats obtenus

5.1.1 Cadre expérimental

Cette section présente la série d'étapes utilisée pour obtenir les résultats présentés aux sous-sections 5.1.3 et 5.1.4. Le but étant de présenter la séquence d'opérations appliquée et de faciliter la reproductibilité des résultats obtenus pour des travaux futurs. Voici donc cette série d'étapes :

1. D'abord, créer trois groupes de classification TEST en utilisant un filtre différent pour chacun d'eux (Aucun, Sobel, Prewitt), ainsi que tous les ensembles d'images d'entraînement (donne trois groupes de classification de 21 catégories).
2. Ensuite, exécuter la classification des ensembles d'images test en utilisant la solution 1 (avec discrimination), puis la solution 2 (sans discrimination) avec les trois groupes de classification¹. Pour chaque groupe testé, les valeurs de la précision de la classification et du temps de traitement obtenues ont été compilées et sont présentées dans les tableaux A.1 et B.1 (voir les annexes A et B).
3. L'étape suivante consiste à déterminer quelle est la catégorie à retirer dans les trois groupes créés à l'étape 1 (le but étant de comparer la précision de la classification et le temps de traitement en fonction du nombre de catégories à classer). Le choix de la catégorie à exclure est fait en fonction des critères suivants :
 - Est-ce la catégorie qui diminue le plus la précision du groupe de classification ?
 - Présente-t-elle une forte similarité avec d'autres catégories (difficilement dissociable) ?
 - Comporte de nombreuses disparités dans les caractéristiques des images d'entraînement ? *Par exemple, les intervalles de certaines caractéristiques couvrent*

¹Pour les deux solutions, ce sont seulement les ensembles d'images appartenant aux catégories des trois groupes de classification qui ont été testés.

l'ensemble des caractéristiques similaires des autres catégories du groupe de classification.

4. Une fois la catégorie à retirer X est ciblée, une copie des trois groupes précédemment créés de N catégories est réalisée, puis la catégorie X est retirée dans chacun de ces nouveaux groupes. On obtient alors trois groupes de $(N - 1)$ catégories.
5. Répéter la séquence d'étapes à partir du point 2 avec les trois groupes nouvellement créés.

Note : Dans le cadre de cette recherche, ce sont des groupes de 21 à 9 catégories qui ont été créées.

Le tableau 5.1 présente les catégories qui ont été identifiées puis retirées des groupes suite aux passages dans la séquence d'étapes ci-haut mentionnées.

ID	Nom de la catégorie	Liste de catégories à classer par groupe ($X = Retirée$)												
		0	1	2	3	4	5	6	7	8	9	10	11	12
1	Asterionellopsis					X	X	X	X	X	X	X	X	X
2	Chaetoceros							X	X	X	X	X	X	X
3	Ciliate										X	X	X	X
4	Cylindrotheca													
5	Cerataulina													
6	Dactyliosolen													
7	Detritus		X	X	X	X	X	X	X	X	X	X	X	X
8	Dinobryon													X
9	Dinoflagellates													
10	Ditylum												X	X
11	Euglena											X	X	X
12	Guinardia								X	X	X	X	X	X
13	Licmophora													
14	Nanoflagellates													
15	Pennate									X	X	X	X	X
16	Phaeocystis													
17	Pleurosigma													
18	Pseudonitzschia				X	X	X	X	X	X	X	X	X	X
19	Rhizosolenia													
20	Skeletonema						X	X	X	X	X	X	X	X
21	Thalassiosira			X	X	X	X	X	X	X	X	X	X	X
Nombre de catégories par groupe		21	20	19	18	17	16	15	14	13	12	11	10	9

Tableau 5.1 Liste des catégories appartenant aux différents groupes de classification TEST

Les sections suivantes présentent la compilation des résultats obtenus pour la précision de la classification, ainsi que du nombre d'images traitées par seconde pour chacune des deux solutions proposées, en utilisant tous les groupes de classification TEST créés.

5.1.2 Spécification du système informatique

Avant de faire la présentation des résultats obtenus, il est important de montrer quels sont les caractéristiques spécifiques du système informatique sur lequel le logiciel *Biotaxis* a été exécuté. La liste des caractéristiques de ce système est :

- Système d'exploitation : Windows 7.
- Mémoire vive : $2 \times$ DDR 1600 MHz de 4 Go.
- Processeur : Intel Core i7-2600.
- Disque dur : Seagate Barracuda 6-Gb/s 500 GB.

5.1.3 Solution 1 - Classification AVEC discrimination

Nombre de catégories à classer	Nombre d'images testées	Précision de la classification (%) <i>Selon le filtre utilisé</i>			Images traitées (par seconde) <i>Selon le filtre utilisé</i>		
		Aucun	Sobel	Prewitt	Aucun	Sobel	Prewitt
21	3138	62.3 %	59.0 %	59.7 %	25.7	25.3	25.8
20	2990	65.9 %	62.0 %	62.8 %	28.0	27.6	27.9
19	2840	68.2 %	65.0 %	65.6 %	29.3	28.8	29.4
18	2690	69.5 %	68.3 %	68.5 %	30.5	30.6	31.0
17	2540	72.0 %	71.1 %	71.5 %	32.3	31.2	31.6
16	2390	73.9 %	73.5 %	74.1 %	34.9	34.2	34.9
15	2240	76.8 %	76.1 %	76.6 %	37.0	36.3	37.3
14	2090	80.6 %	78.9 %	78.4 %	42.1	40.6	41.3
13	1941	83.5 %	81.1 %	80.5 %	45.5	42.5	44.0
12	1791	85.5 %	83.8 %	83.2 %	48.4	43.6	46.3
11	1645	87.8 %	85.2 %	85.1 %	49.7	44.7	46.9
10	1495	88.8 %	87.8 %	88.5 %	52.0	46.5	48.6
9	1345	90.1 %	88.7 %	90.9 %	49.0	42.7	44.8

Tableau 5.2 Résultats obtenus avec la solution 1 selon le filtre utilisé (précision et temps de traitement)

La séquence de traitements utilisée pour la solution 1 ayant permis d'obtenir le tableau des résultats A.1 est :

1. Démarrage du compteur de temps.

2. Caractérisation de l'image selon le filtre.
3. Discrimination de catégories (avec arbre à intervalles).
4. Génération du modèle SVM avec les catégories non discriminées.
5. Classification de l'image (méthode SVM).
6. Validation de la précision de la classification (%).
7. Arrêt du compteur de temps.
8. Calcul de temps de traitement requis pour la classification.

Cette séquence a été exécutée en boucle avec **seulement**² les groupes d'images test pré-classifiés dont les catégories sont définies par les groupes de classification. La compilation des résultats obtenus par image est présentée dans le tableau A.1.

5.1.4 Solution 2 - Classification SANS discrimination

Nombre de catégories à classer	Nombre d'images testées	Précision de la classification (%) <i>Selon le filtre utilisé</i>			Images traitées (par seconde) <i>Selon le filtre utilisé</i>		
		Aucun	Sobel	Prewitt	Aucun	Sobel	Prewitt
21	3138	72.0 %	68.0 %	67.8 %	65.5	60.8	62.2
20	2990	74.9 %	70.8 %	70.9 %	80.9	66.4	66.6
19	2840	76.8 %	72.7 %	73.1 %	76.2	62.1	62.3
18	2690	77.9 %	75.1 %	75.5 %	76.1	62.1	60.9
17	2540	80.5 %	77.7 %	77.9 %	79.5	67.3	66.8
16	2390	82.1 %	79.8 %	79.9 %	78.9	66.5	66.1
15	2240	84.2 %	82.9 %	83.2 %	68.3	67.0	68.6
14	2090	86.0 %	84.4 %	84.6 %	80.0	70.7	69.5
13	1941	88.8 %	86.7 %	87.4 %	79.4	67.9	68.6
12	1791	89.9 %	89.2 %	90.1 %	77.6	67.2	66.4
11	1645	92.0 %	90.9 %	91.4 %	76.6	63.7	65.1
10	1495	93.1 %	92.0 %	92.2 %	73.1	64.7	63.4
9	1345	94.3 %	93.3 %	93.4 %	73.9	56.2	62.4

Tableau 5.3 Résultats obtenus pour la solution 2 selon le filtre utilisé (précision et temps de traitement)

La séquence de traitements utilisée pour la solution 2 ayant permis d'obtenir le tableau des résultats B.1 est :

²La section 6.2 présente une alternative possible à la classification de seulement les images définies par un groupe de classification.

1. Démarrage du compteur de temps.
2. Caractérisation de l'image selon le filtre.
3. Classification de l'image avec la méthode SVM, en employant le modèle SVM généré lors de l'entraînement du groupe de classification TEST utilisé ³.
4. Validation de la précision de la classification (%).
5. Arrêt du compteur de temps.
6. Calcul de temps de traitement requis pour la classification.

Comme à la section précédente (solution 1), la séquence ci-haut a été exécutée en boucle avec **seulement**⁴ les groupes d'images test préclassifiés dont les catégories sont présentes dans les groupes de classification. La compilation des résultats obtenus par image est présentée dans le tableau B.1.

5.2 Analyse des résultats

L'analyse des résultats obtenus, qui est énoncée dans cette section, utilise comme base comparative, deux des objectifs ciblés par ce projet. Ceux-ci sont d'obtenir :

- Une précision moyenne de la classification de 87 %.
- Un temps de traitement moyen de 200 ms par image. Ce qui représente le traitement d'approximativement 5 images traitées par seconde.

En fonction de ces deux objectifs, une analyse de la qualité des résultats obtenus et un comparatif des résultats avec les objectifs (ci-haut mentionnés) a été réalisée. Les sous-sections suivantes présentent une comparaison et une interprétation des résultats obtenus.

5.2.1 Comparaison des résultats des deux solutions

Pour être en mesure d'effectuer une comparaison des résultats entre les deux solutions de classification proposées dans cette recherche, deux graphiques de comparaison ont été générés à partir du jumelage des données provenant des tableaux A.1 et B.1. Ainsi, les résultats de la précision de la classification par filtre de la solution 1 ont été jumelés avec ceux de la solution 2 et la même opération a été effectuée pour les résultats du nombre

³Le modèle SVM est généré avec l'ensemble des catégories possibles du groupe (sans discrimination).

⁴La section 6.2 présente une alternative possible à la classification de seulement les images définies par un groupe de classification.

d'images traitées par seconde selon le filtre utilisé. Les deux graphiques de comparaison sont présentés ci-après.

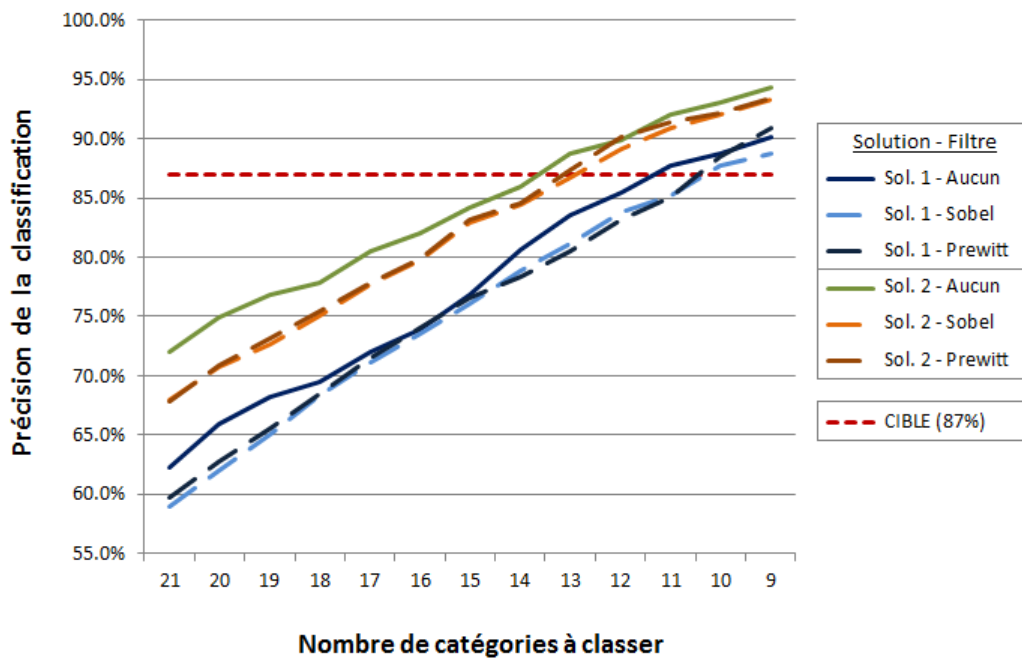


Figure 5.1 Comparaison de la précision de la classification atteinte en fonction du nombre de catégories à classer (par solution et filtre)

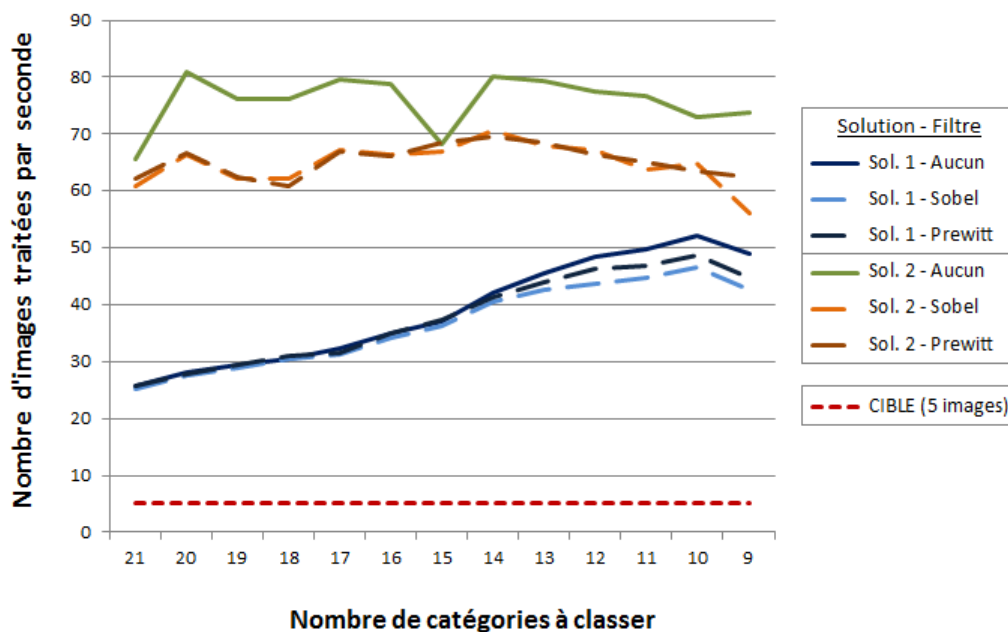


Figure 5.2 Comparaison du nombre d'images traitées par seconde en fonction du nombre de catégories à classer (par solution et filtre)

5.2.2 Interprétation des résultats

Précision de la classification moyenne

En premier lieu, l'analyse du graphique de la figure 5.1 permet de conclure que certains groupes de classification ont réussi à atteindre et même surpasser la cible de précision de la classification de 87 %. Après l'analyse des courbes de ce graphique, on constate que la diminution du nombre de catégories définies par un groupe résulte en une augmentation de la précision de la classification moyenne de ce groupe. Pour cette recherche, les résultats obtenus démontrent qu'il fut possible d'atteindre la cible de précision de la classification moyenne de 87 % avec des groupes de classification définissant 13 catégories avec la solution 2 et 11 catégories avec la solution 1. Les deux solutions ont permis d'atteindre la cible, avec les groupes d'images utilisés, mais il apparaît qu'avec ce jeu d'images, c'est la solution 2 qui est préférable à la solution 1.

De plus, les courbes du graphique de la figure 5.1 indiquent qu'il est possible d'obtenir une précision de la classification moyenne très élevée (supérieure à 90 %), dépendamment du nombre de catégories définies par le groupe de classification testé. Donc plus le nombre de catégories définies par un groupe de classification est bas, plus la valeur de précision de la classification escomptée est élevée. Bien entendu, cela suppose que les approches présentées, et qui ont servi à l'obtention des résultats analysés, sont respectées.

Si l'on compare ces résultats avec les données présentées dans le tableau 2.1⁵, on constate que les deux solutions proposées permettent d'obtenir une précision de la classification supérieure à quatre des six articles énoncés. Cependant, les résultats dans ce tableau ne peuvent pas être reliés à un banc d'essai, car il y a trop d'inconnu quant au nombre de caractéristiques utilisées par les classificateurs, ainsi que la suite de procédés employés. C'est pourquoi il est difficile de comparer adéquatement nos solutions, ainsi que les résultats obtenus, avec ceux présentés dans le tableau 2.1.

Nombre moyen d'images traitées par seconde

L'analyse du second graphique, présentant une comparaison entre le nombre d'images traitées par seconde versus le nombre de catégories à classer, identifie clairement que dans le contexte actuel, où le nombre de catégories à classer ne dépasse pas 21, qu'il est possible de traiter beaucoup plus que 5 images par seconde avec le logiciel *Biotaxis* (voir le tracé de la cible sur le graphique). Peu importe la solution et le filtre utilisés, il est toujours

⁵ Tableau 2.1 : Précision de la classification d'images de phytoplancton obtenue avec l'usage de la méthode SVM selon les articles à l'étude ayant obtenus plus de 70 % de précision de la classification

possible de traiter plus de 5 images par seconde (soit jusqu'à 79 images par seconde selon le cas).

L'analyse des courbes du graphique de la figure 5.2 permet de constater qu'à quelques reprises, certaines courbes présentent une baisse du nombre d'images traitées par seconde et cette baisse est en fait attribuable à la dimension des images utilisées. Effectivement, le temps de traitement d'une image dépend de sa dimension, car plus l'image est grande, plus long sera le temps requis pour effectuer le prétraitement de cette image. Puisque certaines méthodes du prétraitement nécessitent un balayage de l'image, ce balayage sera beaucoup plus long si une image est composée par exemple de 200 000 pixels versus 10 000 pixels.

Les résultats présentés dans la figure 5.2, proviennent de la compilation des temps de traitement moyens globaux des groupes de classification TEST, selon le nombre de catégories définies pour chacun de ces groupes de classification TEST. Alors si un groupe de classification définit un nombre N de catégories ayant des images de grande dimension, il est facile de prévoir que son temps de traitement moyen sera supérieur à celui d'un autre groupe de N catégories ayant des images de petite dimension.

En résumé

Suite à cette analyse, on peut conclure que les deux solutions de classification proposées permettent toutes deux d'atteindre et même dépasser les deux objectifs du projet. Parmi ces deux solutions, il apparaît que la solution 2 (sans discrimination) est celle devant être priorisée dans le contexte actuel, soit de l'usage des groupes d'images de phytoplancton fournis par le WHOI.

CHAPITRE 6

DISCUSSION

Cette partie du mémoire de recherche est la rétrospection de ce qui a bien été lors de la réalisation du logiciel *Biotaxis* (les bonnes décisions) et ce qui devrait être changé ou amélioré (les décisions à reconsidérer).

6.1 Les bonnes décisions

Pour ce projet de recherche, les bonnes décisions ont été multiples. Celles ayant eues le plus d'incidence positive sont les suivantes :

1. D'abord, le choix d'utiliser le langage Java pour la programmation du logiciel a permis d'obtenir un code portable (grâce à l'usage de la JVM¹) et de diminuer le temps de développement, grâce à la réutilisation des nombreuses bibliothèques offertes avec ce langage (tel que ; *jlibsvm*, *Swing*² et *JNI*).
2. Ensuite, le fait d'utiliser la bibliothèque externe *jlibsvm* a facilité l'intégration de la méthode SVM au projet et évité de devoir implémenter les fonctionnalités requises pour cette méthode de traitement.
3. L'ajout de la fenêtre d'analyse du prétraitement fut également une bonne décision. Le fait de pouvoir faire des tests d'intégration et de validation des différents traitements de cette phase via cette fenêtre a grandement contribué à la qualité du résultat de l'étape de prétraitement.
4. Une autre bonne décision fut d'utiliser les groupes d'images préclassifiés fournis par le WHOI. Cet ajout au projet a permis de vérifier les résultats présentés dans ce mémoire de recherche.
5. Finalement, la conception et la réalisation du logiciel de classification *Biotaxis* ouvre la porte à d'autres projets possibles, en lien avec cette recherche (un aperçu des travaux futurs possibles est présenté à la section 7.3).

¹Java Virtual Machine

²Swing est le nom d'une bibliothèque graphique du langage Java.

6.2 Les décisions à considérer

Pour ce qui est des décisions à considérer, celles ayant attirées l'attention sont :

1. Envisager l'ajout, au logiciel *Biotaxis*, d'une option de configuration du vote minimal requis pour effectuer une prédiction de classification d'image avec la méthode SVM. Cette option permettrait de paramétrer la librairie *jlsvm*, afin d'établir une valeur minimale requise pour attribuer une catégorie d'appartenance à une image. Cette ajout pourrait permettre de :
 - (a) Tester la classification d'ensemble d'images avec des groupes de classification ne comprenant pas la définition de toutes les catégories d'images testées.
 - (b) Permettre au logiciel de classifier des images en temps réel, et ce, indépendamment des catégories d'images fournies.
 - (c) Cibler les images n'ayant pas atteint un vote minimal et éventuellement, les sauvegarder sous forme de fichier dans un dossier servant à la classification manuelle.
2. Remettre en question l'utilisation de deux langages de programmation pour la réalisation du logiciel *Biotaxis* (soit le Java et le C++). Ce questionnement sur cette décision est basé sur les points suivants :
 - (a) D'abord, l'ajout de lourdeur au projet dû à la transition d'informations entre le module en Java et celui en C++.
 - (b) La complexité d'intégration et de configuration de la librairie JNI.
 - (c) Le temps de développement beaucoup plus long en langage C++ qu'en Java.
 - (d) L'incertitude quant au gain en temps de traitement réalisé dans ce projet, pour les différents traitements ayant été effectués dans le module en C++. Une proposition à considérer serait de convertir, en langage Java, les modules de chargement d'images et de prétraitement qui sont implémentés en C++, puis de retirer du logiciel les modules utilisant la librairie JNI. Effectuer ces changements permettrait d'avoir un logiciel entièrement développé en langage Java.
3. Vérifier quels seraient les impacts occasionnés par l'ajout de caractéristiques. Certaines interrogations à ce sujet sont à considérer :
 - (a) Est-ce que ces ajouts permettraient de dissocier certaines catégories ayant plusieurs similitudes ?

- (b) Quel serait l'impact sur le temps de traitement ?
- (c) La précision de la classification des catégories serait-elles supérieure ?
- (d) Finalement, quel serait l'impact de ces ajouts sur la précision globale d'un groupe de classification ? Serait-ce alors possible d'avoir un groupe comprenant plus de 13 catégories et ayant une précision de la classification globale, supérieure à 87 % ?

CHAPITRE 7

CONCLUSION

7.1 Récapitulatif

Ce projet de recherche avait pour but de faire une étude de la précision et de la performance du processus de classification d'images de phytoplancton à l'aide de machine à vecteur de support (SVM). Pour réaliser cette étude, trois objectifs distincts ont été fixés.

- Atteindre une précision moyenne de la classification de 87 %.
- Obtenir un temps de traitement moyen global de 200 ms par image.
- Développer un logiciel de classification d'images.

Suite à l'identification de ces objectifs, une présentation des concepts généraux de la classification d'images de phytoplancton fut réalisée au chapitre 2, suivi par une synthèse des avancées sur le sujet dans le chapitre 3 de l'« État de l'art ». Dans ces deux chapitres, deux solutions de classification d'images ont été énoncées. La première étant d'effectuer une classification d'images en utilisant les phases de traitement suivantes : 1 - Prétraitement, 2 - Discrimination de catégorie, 3 - Classification avec la méthode SVM. La seconde solution proposée effectue une classification d'images à partir des phases suivantes : 1 - Prétraitement, 2 - Classification avec la méthode SVM.

Pour être en mesure de tester et valider les deux solutions proposées, un logiciel de classification d'images nommé *Biotaxis* a été développé. Une description de l'analyse, de la création et de l'utilisation de ce logiciel est présentée dans le chapitre 4, portant sur le « Développement ». À partir des groupes d'images fournies par le WHOI, des groupes de classification ont été entraînés et testés à l'aide du logiciel *Biotaxis*. Les résultats de précision de la classification et de temps de traitement obtenus ont été présentés dans le chapitre 5 « Résultats et Analyse ».

L'analyse des résultats obtenus et présentés au chapitre 5 a permis de déterminer que la solution 2 proposée, soit celle n'utilisant pas de phase de discrimination de catégories, était beaucoup plus rapide et précise que la solution 1 (avec discrimination). Également, il fut observé, avec le jeu d'images utilisées, que dépendamment des catégories d'images à classer, qu'il est possible d'obtenir une précision de la classification supérieure à 87 %

avec 13 catégories de phytoplancton différentes, et ce, avec un temps de traitement moyen par image de 12.6 ms (représentant plus de 79 images traitées par seconde).

Ces résultats justifient l’usage des différentes approches présentées dans cette recherche. Ayant atteint les deux objectifs : soit d’obtenir une précision de la classification moyenne de 87 % et un temps de traitement moyen de 200 ms par image. Il est donc possible de partir de cette base et de cibler des objectifs de précision et de temps de traitement encore meilleurs.

7.2 Contributions

Nous proposons d’abord le logiciel de classification *Biotaxis* comme étant une nouvelle solution pour classifier rapidement des images de phytoplancton. Grâce aux différentes options de traitement intégrées au logiciel, il est possible de :

- Paramétrer le prétraitement d’une image,
- Visualiser les traitements effectués sur une image lors de l’étape de prétraitement.
- Utiliser deux méthodes de classification (avec et sans discrimination de catégories).
- Consulter les différents résultats des tests de classification, d’expérimenter l’usage de différents groupes de classification et de comparer les résultats obtenus.
- Effectuer des choix éclairés quant aux images à utiliser pour l’entraînement d’un groupe de classification, puisqu’il est possible de vérifier la précision de la classification du groupe selon les images choisies.

En second lieu, nous présentons deux solutions permettant la classification d’images. Ces deux solutions, ayant été intégrées au logiciel *Biotaxis*, offrent à l’utilisateur du logiciel, l’option de choisir le type de classification désiré. L’avantage étant de pouvoir réaliser un comparatif entre les approches utilisées et de déterminer laquelle est la plus optimale selon le contexte.

Le choix revient à l’utilisateur de configurer le logiciel, afin d’orienter son mode de classification en fonction des résultats recherchés.

7.3 Travaux futurs

Parmi les travaux futurs pouvant être dérivés du projet *Biotaxis*, il serait intéressant de :

- Tester l'application logicielle avec un groupe d'images autre que celles appartenant aux phytoplanctons. Avec l'abondance de la diversité biologique terrestre, plusieurs types d'images pourraient être envisagées.
- Intégrer au logiciel le traitement d'images via un processeur graphique (GPU¹). Il pourrait-être considéré d'ajouter une option d'activation du traitement d'images en mode GPU. Lorsqu'activé, certaines ou l'ensemble des séquences de traitements d'image seraient alors effectuées sur le GPU. L'intégration au logiciel *Biotaxis* de processus effectués sur un GPU pourrait avoir un impact positif sur le temps de traitement requis pour classer une image.
- Ajouter un module de lecture de « script » au logiciel, visant à faciliter l'ajout de méthodes de génération de caractéristiques. Par exemple, des usagers pourraient-être intéressés par la caractérisation de :
 - la texture dans une image ;
 - la présence de noyaux ;
 - la variation de l'intensité lumineuse.

Grâce à ce module, des utilisateurs pourraient intégrer au logiciel des méthodes de caractérisation d'images, sans qu'ils aient au préalable, recours à des connaissances en programmation.

¹GPU : Graphics processing unit

ANNEXE A

DONNÉES

Essai	Nombre de Catégories	Filtre	Nombre d'images	Précision (%)	VP	FP	Temps de traitement (sec)		Images traitées (par seconde)
							Total	Par image	
1	21	Aucun	3138	62.3	1954	1184	122.076	0.0389	25.7
2	21	Sobel	3138	59.0	1851	1287	123.830	0.0395	25.3
3	21	Prewitt	3138	59.7	1872	1266	121.512	0.0387	25.8
4	20	Aucun	2990	65.9	1956	1034	106.729	0.0357	28.0
5	20	Sobel	2990	62.0	1846	1137	108.227	0.0362	27.6
6	20	Prewitt	2990	62.8	1861	1111	107.250	0.0359	27.9
7	19	Aucun	2840	68.2	1936	904	97.035	0.0342	29.3
8	19	Sobel	2840	65.0	1846	994	98.624	0.0347	28.8
9	19	Prewitt	2840	65.6	1864	976	96.456	0.0340	29.4
10	18	Aucun	2690	69.5	1869	821	88.318	0.0328	30.5
11	18	Sobel	2690	68.3	1837	853	87.840	0.0327	30.6
12	18	Prewitt	2690	68.5	1842	848	86.643	0.0322	31.0
13	17	Aucun	2540	72.0	1829	711	78.576	0.0309	32.3
14	17	Sobel	2540	71.1	1807	733	81.475	0.0321	31.2
15	17	Prewitt	2540	71.5	1815	725	80.458	0.0317	31.6
16	16	Aucun	2390	73.9	1767	623	68.463	0.0286	34.9
17	16	Sobel	2390	73.5	1757	633	69.976	0.0293	34.2
18	16	Prewitt	2390	74.1	1770	620	68.567	0.0287	34.9
19	15	Aucun	2240	76.8	1719	520	60.571	0.0270	37.0
20	15	Sobel	2240	76.1	1704	536	61.665	0.0275	36.3
21	15	Prewitt	2240	76.6	1715	525	60.120	0.0268	37.3
22	14	Aucun	2090	80.6	1684	405	49.680	0.0238	42.1
23	14	Sobel	2090	78.9	1648	442	51.455	0.0246	40.6
24	14	Prewitt	2090	78.4	1639	451	50.602	0.0242	41.3
25	13	Aucun	1941	83.5	1620	319	42.633	0.0220	45.5
26	13	Sobel	1941	81.1	1575	366	45.636	0.0235	42.5
27	13	Prewitt	1941	80.5	1563	378	44.129	0.0227	44.0
28	12	Aucun	1791	85.5	1531	260	37.013	0.0207	48.4
29	12	Sobel	1791	83.8	1501	290	41.046	0.0229	43.6
30	12	Prewitt	1791	83.2	1491	300	38.688	0.0216	46.3
31	11	Aucun	1645	87.8	1444	201	33.129	0.0201	49.7
32	11	Sobel	1645	85.2	1401	244	36.760	0.0223	44.7
33	11	Prewitt	1645	85.1	1400	245	35.040	0.0213	46.9
34	10	Aucun	1495	88.8	1327	168	28.767	0.0192	52.0
35	10	Sobel	1495	87.8	1312	183	32.154	0.0215	46.5
36	10	Prewitt	1495	88.5	1323	172	30.781	0.0206	48.6
37	9	Aucun	1345	90.1	1212	133	27.469	0.0204	49.0
38	9	Sobel	1345	88.7	1193	152	31.523	0.0234	42.7
39	9	Prewitt	1345	90.9	1223	122	29.992	0.0223	44.8

Tableau A.1 Resultats obtenus avec la solution 1 (AVEC discrimination)

ANNEXE B

DONNÉES (Suite)

Essai	Nombre de Catégories	Filtre	Nombre d'images	Précision (%)	VP	FP	Temps de traitement (sec)		Images traitées (par seconde)
							Total	Par image	
1	21	Aucun	3138	72.0	2259	879	47.894	0.0153	65.5
2	21	Sobel	3138	68.0	2133	1005	51.576	0.0164	60.8
3	21	Prewitt	3138	67.8	2129	1009	50.428	0.0161	62.2
4	20	Aucun	2990	74.9	2241	749	36.940	0.0124	80.9
5	20	Sobel	2990	70.8	2117	873	45.033	0.0151	66.4
6	20	Prewitt	2990	70.9	2121	869	44.925	0.0150	66.6
7	19	Aucun	2840	76.8	2182	658	37.260	0.0131	76.2
8	19	Sobel	2840	72.7	2064	776	45.703	0.0161	62.1
9	19	Prewitt	2840	73.1	2075	765	45.621	0.0161	62.3
10	18	Aucun	2690	77.9	2095	595	35.351	0.0131	76.1
11	18	Sobel	2690	75.1	2021	669	43.333	0.0161	62.1
12	18	Prewitt	2690	75.5	2031	659	44.177	0.0164	60.9
13	17	Aucun	2540	80.5	2044	496	31.948	0.0126	79.5
14	17	Sobel	2540	77.7	1973	567	37.746	0.0149	67.3
15	17	Prewitt	2540	77.9	1978	562	38.011	0.0150	66.8
16	16	Aucun	2390	82.1	1963	427	30.308	0.0127	78.9
17	16	Sobel	2390	79.8	1908	482	35.956	0.0150	66.5
18	16	Prewitt	2390	79.9	1910	480	36.183	0.0151	66.1
19	15	Aucun	2240	84.2	1887	353	32.774	0.0146	68.3
20	15	Sobel	2240	82.9	1856	384	33.445	0.0149	67.0
21	15	Prewitt	2240	83.2	1864	376	32.662	0.0146	68.6
22	14	Aucun	2090	86.0	1797	293	26.140	0.0125	80.0
23	14	Sobel	2090	84.4	1763	327	29.581	0.0142	70.7
24	14	Prewitt	2090	84.6	1769	321	30.066	0.0144	69.5
25	13	Aucun	1941	88.8	1724	217	24.448	0.0126	79.4
26	13	Sobel	1941	86.7	1682	259	28.592	0.0147	67.9
27	13	Prewitt	1941	87.4	1697	244	28.300	0.0146	68.6
28	12	Aucun	1791	89.9	1611	180	23.080	0.0129	77.6
29	12	Sobel	1791	89.2	1598	193	26.654	0.0149	67.2
30	12	Prewitt	1791	90.1	1614	177	26.974	0.0151	66.4
31	11	Aucun	1645	92.0	1513	132	21.487	0.0131	76.6
32	11	Sobel	1645	90.9	1496	149	25.818	0.0157	63.7
33	11	Prewitt	1645	91.4	1503	142	25.260	0.0154	65.1
34	10	Aucun	1495	93.1	1392	103	20.452	0.0137	73.1
35	10	Sobel	1495	92.0	1375	120	23.103	0.0155	64.7
36	10	Prewitt	1495	92.2	1378	117	23.578	0.0158	63.4
37	9	Aucun	1345	94.3	1269	76	18.192	0.0135	73.9
38	9	Sobel	1345	93.3	1255	90	23.912	0.0178	56.2
39	9	Prewitt	1345	93.4	1256	89	21.547	0.0160	62.4

Tableau B.1 Resultats obtenus avec la solution 2 (SANS discrimination)

ANNEXE C

Analyse des métriques du logiciel *Biotaxis*

À noter que l'avertissement pour l'usage d'une fonction avec un nombre de paramètres trop élevé (soit 8), est dû à l'usage d'une méthode provenant d'une librairie de Java via une surcharge (« @Override »).

Metric	Total	Mean	Std. Dev.	Max...	Resource caus...	Method
▷ Number of Overridden Methods (avg/max per	58	0.492	1.24	5	/biotaxis/Viewer...	
▷ Number of Attributes (avg/max per type)	873	7.398	8.1	54	/biotaxis/Viewer...	
▷ Number of Children (avg/max per type)	0	0	0	0	/biotaxis/Viewer...	
▷ Number of Classes (avg/max per packageFrag	118	8.429	7.66	26	/biotaxis/Viewer...	
▷ Method Lines of Code (avg/max per method)	11650	8.153	13.262	181	/biotaxis/Viewer...	initAllListener
▷ Number of Methods (avg/max per type)	1413	11.975	11.072	63	/biotaxis/Viewer...	
▷ Nested Block Depth (avg/max per method)		1.445	0.856	6	/biotaxis/Viewer...	removeCategoryInGroup
▷ Depth of Inheritance Tree (avg/max per type)		2.831	1.815	6	/biotaxis/Viewer...	
▷ Number of Packages	14					
▷ Affrent Coupling (avg/max per packageFragm		14.286	21.039	74	/biotaxis/Viewer...	
▷ Number of Interfaces (avg/max per packageFri	22	1.571	3.54	14	/biotaxis/Viewer...	
▷ McCabe Cyclomatic Complexity (avg/max per		1.796	1.56	10	/biotaxis/Viewer...	editBioGroup
▷ Total Lines of Code	19381					
▷ Instability (avg/max per packageFragment)		0.574	0.336	1	/biotaxis/Viewer...	
▷ Number of Parameters (avg/max per method)		0.924	1.107	8	/biotaxis/Viewer...	paintTabBackground
▷ Lack of Cohesion of Methods (avg/max per typ		0.606	0.329	0.961	/biotaxis/Viewer...	
▷ Effrent Coupling (avg/max per packageFragm		7.357	5.273	15	/biotaxis/Viewer...	
▷ Number of Static Methods (avg/max per type)	17	0.144	0.586	4	/biotaxis/Viewer...	
▷ Normalized Distance (avg/max per packageFra		0.338	0.341	1	/biotaxis/Viewer...	
▷ Abstractness (avg/max per packageFragment)		0.121	0.238	0.933	/biotaxis/Viewer...	
▷ Specialization Index (avg/max per type)		0.155	0.397	2	/biotaxis/Viewer...	
▷ Weighted methods per Class (avg/max per typ	2567	21.754	22.302	108	/biotaxis/Viewer...	
▷ Number of Static Attributes (avg/max per type	182	1.542	3.793	31	/biotaxis/Viewer...	

Figure C.1 Métriques du module en Java de *Biotaxis* (le 19 mai 2014)

LISTE DES RÉFÉRENCES

- Abe, S. (2010). *Support Vector Machines for pattern classification - Second Edition*. Springer, 1-471 p.
- Alvarez, E., Lopez-Urrutia, A. et Nogueira, E. (2012). Improvement of plankton biovolume estimates derived from image-based automatic sampling devices : application to flowcam. *Journal of plankton research*, volume 34, numéro 6, p. 454–469.
- Baillargeon, G. (1990). *Méthodes statistiques de l'ingénieur, volume 1 (3e édition)*. Les Éditions SMG, 313-366 p.
- Bates, B., Robson, E. et Freeman, E. (2004). *Head First Design Patterns*. O'Reilly Media, 1-678 p.
- Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 6, p. 679–698.
- Chang, C.-C. et Lin, C.-J. (2011). LIBSVM : A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, volume 2, p. 27 :1–27 :27.
- Cormen, T. H., Leiserson, C. E. et Rivest, R. L. (1989). *Introduction to algorithms*. McGraw-Hill Book Company, 290-294 p.
- Cover, T. M. et Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, volume 13, numéro 1, p. 21–27.
- de Berg, M., Cheong, M., van Kreveld, M. et Overmars, M. (2008). *Computational Geometry : Algorithms and Applications (Third Edition)*. Springer, 2-8 p.
- Eclipse (2013). *Kepler*. <http://www.eclipse.org/> (page consultée le 7 août 2013).
- Encyclopedia, O. (Octobre 2011). *biotaxis*. <http://www.encyclo.co.uk/define/biotaxis> (page consultée le 30 juillet 2012).
- Falkowski, P. G. et Raven, J. A. (2007). *Aquatic photosynthesis, 2nd Edition*. Blackwell Publishers, 1-375 p.
- Fawcett, T. (2006). Roc analysis in pattern recognition. *Pattern Recognition Letters*, volume 27, numéro 8, p. 861–874.
- Gonzalez, R. C., Woods, R. E. et Eddins, S. L. (2009). *Digital Image Processing Using Matlab - 2nd Edition*. Gatesmark Publishing, 1-827 p.
- Hamel, L. (2009). *Knowledge discovery with Support Vector Machine*. Wiley, 1-246 p.
- Hsu, C.-W., Chang, C.-C. et Lin, C.-J. (2010). A practical guide to support vector classification. *Department of Computer Science, National Taiwan University*.
- Hsu, C.-W. et Lin, C.-J. (2002). A comparison of methods for multiclass support vector machines. *Limnology and Oceanography : Methods*, volume 13, numéro 2, p. 415–425.

- Hu, Q. et Davis, C. (2005). Automatic plankton image recognition with co-occurrence matrices and support vector machine. *Marine Ecology Progress Series*, volume 295, p. 21–31.
- Java (2012). *Learn About Java Technology*. <http://www.java.com/en/about/> (page consultée le 14 août 2012).
- Kang, L., Gong, Y., Yang, C., Luo, J., Luo, Q. et Gao, Y. (2010). Marine phytoplankton recognition using hybrid classification methods. *IEEE : 2010 4th International Conference on Bioinformatics and Biomedical Engineering (iCBBE)*, p. 1–5.
- Kovesi, P. (1999). Image features from phase congruency. *Videre : A Journal of Computer Vision Research*. MIT Press, volume 1, numéro 3, p. 1–26.
- Liang, S. (1999). *Java Native Interface : Programmer's Guide and Specification*. Pearson Education, 1-320 p.
- Luo, D. (1998). *Pattern Recognition and image processing, Chap. 3 - Shape Analysis*. Horwood Publishing, 27-83 p.
- Luo, T. et Kramer, K. (2004). Recognizing plankton images from the shadow image particle profiling evaluation recorder. *IEEE Transactions on Systems, Man, and Cybernetics - Part B : Cybernetics*, volume 34, numéro 4, p. 1753–1762.
- Metrics (2014). *Metrics 1.3.6 - Getting started*. <http://metrics.sourceforge.net/> (page consultée le 19 mai 2014).
- Olson, R. J. et Sosik, H. M. (2007). A submersible imaging-in-flow instrument to analyze nano and microplankton : Imaging flowcytobot. *Limnology and Oceanography : Methods*, volume 5, p. 195–203.
- Prewitt, J. M. S. (1970). Object enhancement and extraction. Dans *Picture Processing and Psychopictorics*. B. Lipkin and A. Rosenfeld, Eds. New York : Academic, p. 75–149.
- Pridmore, T. P. (2002). Thresholding images of line drawings with hysteresis. Dans *Graphics Recognition Algorithms and Applications*. Springer, p. 310–319.
- Reddy, M. (2011). *API design for C++*. Morgan Kaufmann Publishers, 1-441 p.
- Rosenblatt, F. (1958). The perceptron : A probabilistic model for information storage and organization in the brain. *Psychological Review*, volume 65, numéro 6, p. 386–408.
- Safavian, S. R. et Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, volume 21, numéro 3, p. 660–674.
- Sobel, I. et Feldman, G. (1968). A 3x3 isotropic gradient operator for image processing. *Stanford Artificial Project (Presentation)*.
- Soergel, D. (2012). *jlibsvm*. <http://dev.davidsoergel.com/trac/jlibsvm/> (page consultée le 16 mai 2014).

- Sosik, H. M. et Olson, R. J. (2007). Automated taxonomic classification of phytoplankton sampled with imaging-in-flow cytometry. *Limnology and Oceanography : Methods*, volume 5, p. 204–216.
- Su, C., Yang, C., Lin, H. et Kang, L. (2010). A system for identification of marine phytoplankton. *IEEE : 2010 2nd International Conference on Signal Processing Systems (ICSPPS)*, volume 3, p. 426–430.
- Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer, 1-188 p.
- Verikas, A., Gelzinis, A., Bacauskiene, M., Olenina, I., Olenin, S. et Vaiciukynas, E. (2012). Phase congruency-based detection of circular objects applied to analysis of phytoplankton images. *Pattern Recognition*, volume 45, numéro 4, p. 1659–1670.